



Numerical simulation of incompressible flows interacting with forced deformable bodies: Application to fish swimming

Seyed Amin Ghaffari

► To cite this version:

Seyed Amin Ghaffari. Numerical simulation of incompressible flows interacting with forced deformable bodies: Application to fish swimming. Fluid mechanics [physics.class-ph]. Aix Marseille université, 2014. English. NNT: . tel-01223248

HAL Id: tel-01223248

<https://hal.science/tel-01223248>

Submitted on 2 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université d'Aix-Marseille - Faculté des Sciences et Techniques
Laboratoire de Mécanique, Modélisation & Procédés Propres (M2P2) UMR 7340 - CNRS

Ecole Doctorale :
“Sciences pour l’Ingénieur : Mécanique, Physique, Micro et Nanoélectronique”

Thèse présentée pour obtenir le grade universitaire de docteur
Discipline : Mécanique et Physique des Fluides

par : Seyed Amin Ghaffari

Simulations numériques d’écoulements incompressibles interagissant avec un corps déformable : application à la nage des poissons

Soutenue le 15 decembre 2014 devant le jury :

Jean-Paul Caltagirone	Professeur, Université de Bordeaux	Rapporteur
Michael Schäfer	Professeur, Université de Darmstadt, Allemagne	Rapporteur
Frédéric Boyer	Professeur, Ecole des Mines de Nantes	Examineur
Michel Visonneau	Directeur de recherche CNRS, Ecole Centrale de Nantes	Examineur
Patrick Bontoux	Directeur de recherche CNRS, M2P2, Marseille	Examineur
Fabien Candelier	Maître de conférences, IUSTI, Aix-Marseille Université	Invité
Stéphane Viazzo	Maître de conférences, HDR, M2P2, Aix-Marseille Université	Directeur de thèse
Kai Schneider	Professeur, M2P2, Aix-Marseille Université	Co-directeur de thèse

اگر قبرستان جای است که مرگ
 را در آن به خاک سپردہ اند پس ما قبرستان زندگان
 در روز مگر کیا را کے را ہی بہ معنای زندگی است مگر
 مقصد پرواز است ، قفس دیران بہتر از سیر ہے کہ
 مقصد را در کوچ ہے یا بہر از دیرانی لاندہ ہی ہنسی ہر اسد

سید قلی آوی

If cemetery is the area where dead people are interred,
 when I will escape from the cemetery of quotidian life,
 to find the meaning of the real life?

The shorebird how will migrate,
 don't care about his nest,

if the aim is departure,
 a ruinous cottage is better.

Morteza Avini

Remerciements

Je tiens à remercier tous les chercheurs, enseignants, personnels administratifs et techniques de l'université d'Aix-Marseille, de l'Ecole Centrale de Marseille et surtout du laboratoire M2P2 pour m'avoir accueilli dans le cadre du travail de recherche pour préparer ma thèse en mécanique et physique des fluides. En particulier, je souhaiterais remercier les responsables de l'Ecole doctorale ED353 "Sciences pour l'Ingénieur : Mécanique, Physique, Micro et Nanoélectronique", Richard Saurel, Christophe Muller, Eric Serre et Pascal Campion pour leurs efforts lors de mon inscription et durant mes années de thèse. Je remercie également mesdames Corinne Cuoc, Josiane Diaz, Elisabeth Vanni, Karine Wepierre, Virginie Duros et Sylvie-Alexandra Carré pour leurs aides lors des démarches administratives durant mon inscription.

Je souhaite également remercier tous mes professeurs de Master en Mécanique des fluides et physique non-linéaire, organisé par le laboratoire IRPHE, pour m'avoir enseigné des savoirs fondamentaux ainsi que pour m'avoir transmis la culture du travail de recherche. A cette occasion, je dois mentionner Pascale Aussillous, Fabien Anselmet, Paul Clavin, Uwe Ehrenstein, Lazhar Houas, Georges Jourdan, Christian Kharif, Thomas Leweke, Michael Le Bars, Marc Léonetti, Jacques Massoni, Bernard Molin, Fabrice Onofri, Alain Pocheau, Kai Schneider, Yves-Marie Scolas et Emmanuel Villermaux. Je me souviens de la générosité de madame Karine Boudoyan responsable de la bibliothèque d'IRPHE pour m'avoir fourni les ouvrages nécessaires.

J'ai effectué ce travail de recherche au sein du laboratoire de Mécanique, Modélisation et Procédés Propres (M2P2). En premier lieu, je souhaite remercier Patrick Bontoux directeur du laboratoire qui m'a suivi tout au long de mes travaux. Je me souviens des efforts tenaces qu'il a prodigués lors de mon inscription. Ensuite, je dois remercier les personnels de laboratoire, Virginie Blanc, Michel Pognant pour leur aide et Dominique Fougère qui a passé des heures et des heures chaque fois avec mes problèmes particuliers. Je voudrais adresser mes remerciements à tous les chercheurs du M2P2, je pense en particulier à Bernard Roux, Anthony Randriamampianina, Eric Serre, Umberto D'Ortona, Sébastien Poncet, Julien Favier et Denis Martinand. Je ne saurais oublier Stéphane Viazzo et Kai Schneider, mes directeurs de thèse, pour m'avoir fait partager leur expérience et qui ont eu la lourde tâche de diriger ce projet de recherche et pour m'avoir donné les nombreux conseils pendant le déroulement de la thèse. Je ne peux pas oublier les thésards du M2P2, surtout Dimitry Kolomenskiy pour son aide lorsque je suis arrivé au laboratoire.

Je remercie Jean-Paul Caltagirone et Michael Schäfer d'avoir accepté d'être rapporteur de cette thèse ainsi que Frédéric Boyer, Michel Visonneau, Fabien Candelier et Patrick Bontoux en tant qu'examinateur. Leurs commentaires ont beaucoup amélioré la thèse et étaient très utiles pour diriger la suite du travail. Je dois également remercier Margarete Oliveira Domingues (LAC/INPE-Brésil) et Sônia Maria Gomes (Unicamp Campinas-Brésil) pour les discussions qui m'ont été très utiles, Marie Farge (ENS-Paris) pour m'avoir encouragé à apprendre les ondelettes, Frédéric Boyer (Ecole des Mines de Nantes) pour m'avoir expliqué le modèle cinématique concernant la nage tridimensionnelle anguilliforme et envoi de leur code, Mattia Gazzola (Department of Computational Science-Zurich) pour avoir mis à ma disposition les données nécessaires pour les comparaisons et T. B. Gatski éditeur en chef de International Journal of Heat and Fluid Flow.

Je pense aussi à tous mes anciens professeurs et mes amis, je dois citer Feridoun Sabetghadam, Kazem Hejranfar, Karim Mazaheri, Iraj Mirzaee, Majid Abassalizadeh, Payam Sharifi, M. Barandouzi, Mohamad Jafar Hamidia, Alireza Asghari Jazi, Reza Yar Mohammadi, Hossein Norouzi, Mohamad Valizadeh, Iman Farahbakhsh, Amin Paknejad, Abolfazl Moussavinia, Alireza Jafarpour, Mohsen Gorji, Mostafa Ahangharan et Mehdi Sabet qui nous a quitté. Je dois également remercier les responsables de l'école primaire Mahdi, collège Eslami, lycée Dehkhoda et l'université d'Orumieh. Enfin je pense à mes parents ainsi qu'à mon frère pour leurs soutiens. Et je termine par remercier Madame Baba Nejad, qu'elle m'a appris à écrire mon nom en cours préparatoire.

Simulations numériques d'écoulements incompressibles interagissant avec un corps déformable : application à la nage des poissons

Résumé

Une méthode numérique précise et efficace est proposée pour la simulation de corps déformables interagissant avec un écoulement incompressible. Les équations de Navier–Stokes, considérées dans leur formulation vorticité-fonction de courant, sont discrétisées temporellement et spatialement à l'aide respectivement d'un schéma d'ordre quatre de Runge–Kutta et par des différences finies compactes. Grâce à l'utilisation d'un maillage uniforme, nous proposons un nouveau solveur direct au quatrième ordre pour l'équation de Poisson, permettant de garantir la contrainte d'incompressibilité au niveau du zéro machine sur une grille à pas d'espace optimale. L'introduction d'un corps déformable et mobile dans l'écoulement de fluide est réalisée au moyen d'une méthode de pénalisation de volume. La déformation du corps est imposée par l'utilisation d'un maillage lagrangien structuré et mobile qui interagit avec le fluide environnant en raison des forces hydrodynamiques et du moment (calculés sur le maillage eulérien de référence). Une loi efficace de contrôle de la courbure pour un poisson anguilliforme nageant vers un objectif prescrite est proposée. La loi de contrôle de la courbure est basée sur la théorie exacte des poutres non-linéaires. Ensuite pour augmenter l'efficacité de solveur, une analyse multiéchelle est appliquée à l'algorithme, permet de réduire significativement le nombre de points de maillage. La grille se raffine automatiquement dans les régions avec un fort gradient. La stratégie d'adaptation est basée sur la transformée en ondelettes et le seuillage des coefficients. Les résultats obtenus montrent que le temps de calcul peut être réduit considérablement avec la méthode multiéchelle tout en conservant la précision. Finalement une simulation de nage trois-dimensionnel a été faite par méthode de pénalisation de volume appliquée au code Incompact3d qui est en accès libre. La méthode numérique développée prouve son efficacité et précision tant dans le cas de la nage du poisson que dans le cas d'autres problèmes d'interactions fluide–structure.

Mots-clés : Différences finies compactes - Solveurs direct et itératifs - Un corps avec déformation imposée - Pénalisation de volume - Adaptation de maillage par ondelette - Théorie exacte des poutres

Laboratoire M2P2 - UMR 7340, Ecole Centrale Marseille IMT - La Jetée, 38 rue Frédéric Joliot-Curie, Technopôle de Château-Gombert, 13451 Marseille cedex 20 - France

Numerical simulation of incompressible flows interacting with forced deformable bodies: Application to fish swimming

Abstract

We present an efficient algorithm for simulation of deformable bodies interacting with incompressible flows. The temporal and spatial discretizations of the Navier–Stokes equations in vorticity-stream function formulation are based on classical fourth-order Runge–Kutta method and compact finite differences, respectively. Using a uniform Cartesian grid we benefit from the advantage of a new fourth-order direct solver for the Poisson equation to ensure the incompressibility constraint down to machine zero over an optimal grid. For introducing a deformable body in fluid flow, the volume penalization method is used. A Lagrangian structured grid with prescribed motion covers the deformable body which is interacting with the surrounding fluid due to the hydrodynamic forces and the torque calculated on the Eulerian reference grid. An efficient law for controlling the curvature of an anguilliform fish, swimming toward a prescribed goal, is proposed which is based on the geometrically exact theory of nonlinear beams and quaternions. Furthermore to reduce the computational effort, better resolving the boundary layer and the vortical structures, adaptation of grid is performed by using multiresolution analysis. The method is based on Harten’s point value representation, which through nonlinear filtering of the wavelet coefficients reduces the number of active grid points significantly. Finally an extension to three dimensional swimming is performed by adding the implicit volume penalization method to the Incompact3d open access code, to be able to take into account the deformable bodies interaction with incompressible flows. Validation of the developed method shows the efficiency and expected accuracy of the algorithm for fish-like swimming and also for a variety of fluid/solid interaction problems.

Keywords: Compact finite differences - Direct and iterative solvers - Fluid interaction with moving bodies - Volume penalization method - Wavelet based grid adaptation - Geometrically exact beam theory

Notice about nomenclature

This document was produced by L^AT_EX [95]. The attempt was to not use duplicated symbols and superscripts for the same physical quantities or mathematical operations, nevertheless, some prevalent symbols are kept to prevent additional confusion, therefore we tried to introduce each symbol in the used context.

Contents

1	Introduction	1
2	An algorithm for fluid–structure interaction in two dimensions	5
2.1	Vorticity-stream function formulation	6
2.1.1	Volume penalization method	10
2.1.2	Evaluation of the hydrodynamic coefficients	14
2.1.3	Body dynamics	15
2.2	Compact methods	16
2.2.1	Spatial discretization on uniform grids	17
2.2.2	Spatial discretization on non-uniform grids	18
2.2.3	Analysis of differentiation errors	19
2.2.4	Interpolation and filtering	22
2.3	Spatial discretization of the Poisson equation	28
2.3.1	An iterative method for solution of the Poisson equation	29
2.3.2	A high-order fast Poisson solver	34
2.4	Time integration	38
2.5	The algorithm for fluid–structure interaction	42
2.6	Conclusion	42
3	Convergence study and validation of the proposed algorithm	47
3.1	Decaying Taylor vortices	47
3.2	Taylor–Couette flow	49
3.3	Unsteady flow around a circular cylinder	53
3.3.1	Hydrodynamic forces via a control volume around body	55
3.4	Fluid–structure interaction	59
3.4.1	Free sedimentation of a cylinder	59
3.4.2	Validation of the solid dynamics with a falling ellipse	65
3.5	Conclusion	69
4	Two-dimensional simulations of fish-like swimming	76
4.1	Physical definitions	77
4.1.1	Efficiency measurement	82
4.2	Modeling of the swimmer shape	84
4.2.1	Backbone deformation law for straight swimming	85
4.2.2	Kinematics of the fish based on curvature	92
4.2.3	Lagrangian structured grid covering the body	93
4.3	Validation of the algorithm for deformable bodies	96
4.4	Application and results	99

4.5	Conclusion	103
5	Fluid–structure interaction on adaptive grids	112
5.1	General introduction	113
5.2	Multiresolution analysis of incompressible flow	114
5.2.1	Biorthogonal wavelet transform	115
5.2.2	Filtering of wavelet coefficients	117
5.3	Dipole-wall collision with a uniform grid solver	118
5.4	Validation of the adaptive multiresolution solver	122
5.5	Application to fish-like swimming	122
5.6	Conclusion	123
6	Three-dimensional simulations of fish-like swimming	130
6.1	Governing equations	131
6.1.1	Mathematical properties of the N-S equations	132
6.1.2	Conservation properties	132
6.1.3	Grid arrangement	134
6.2	Projection method	137
6.3	Incompact3d code and our modification	140
6.4	Validation of the algorithm for a falling sphere	144
6.5	Three dimensional simulation of swimming fish	145
6.6	Conclusion	160
7	Conclusion and perspectives	165
8	Résumé de thèse en français	168
8.1	Modèle mathématique	169
8.2	Dynamique d’un objet mobile	171
8.3	Discrétisation temporelle et spatiale	173
8.4	Solveur de Poisson rapide	173
8.5	Modèle cinématique de nage anguilliforme	175
8.6	Algorithme d’interaction fluide–structure	180
8.7	Changement de direction du poisson	180
8.8	Conclusion	185
A	Compact differentiation	190
B	The volume penalization method	194
C	The coefficient matrix of Poisson equation	196
D	Fourier transforms	197
E	Turbulent structure identification criteria	199
F	Direct solvers for linear systems with diagonal matrix of coefficients	200
	Bibliography	202

List of Figures

2.1	Domain of the solution and the immersed body, $\Omega = \Omega_f \cup \Omega_p$	11
2.2	(a) Smooth Dirac δ function (2.18) proposed by Lai and Peskin [76]. (b) Comparison of the derivatives of the smoothed mask function computed via the second-order central finite difference method and the fourth-order compact finite difference (Padé) method with the analytical smoothed Dirac delta function (2.18).	14
2.3	One-dimensional stretched grid, $x \in [0, L_x]$, with clustering near boundaries via Eq. (2.38), with $N = 41$, $L_x = 4$, $\gamma_x = 4$ and $\Delta\xi = 1$	19
2.4	Truncation error analysis for first (c-d) and second (e-f) derivatives via a fourth-order compact method for $f(x) = \sin x$, $x \in [0, 2\pi]$, on a uniform grid (a), and $g(x) = (x - 0.5)^4$, $x \in [0, 1]$, on a clustered grid near boundaries (b) with Eq. (2.38), where $\gamma = 4$ and $\beta = \xi_{max}/2$	20
2.5	Truncation error analysis for first (a) and second (b) derivatives via different explicit and compact methods for $f(x) = \sin x$, $x \in [0, 2\pi]$, on a uniform grid.	21
2.6	Plots of the scaled modified wavenumber $w'(w)$ versus the scaled wavenumber $w = k\Delta x$ for the first derivative with the use of different central finite difference methods for $f(x) = e^{ikx}$ and $x \in [0, 2\pi]$	23
2.7	Plots of the resolution error $\epsilon(w) = \frac{w'(w)-w}{w} $ for the first derivative via two pentadiagonal spectral like compact method proposed by Lele [55] and Kim [111], with $N = 2^{10}$ grid points.	23
2.8	Plots of the scaled modified wavenumber $w''(w)$ versus the scaled wavenumber $w = k\Delta x$ for the second derivative with the use of different central finite difference methods for $f(x) = e^{ikx}$ and $x \in [0, 2\pi]$	24
2.9	Plots of the resolution error $\epsilon(w) = \frac{w''(w)-w^2}{w^2} $, for the second derivative via different methods.	25
2.10	Transfer functions $T(w)$ versus scaled wavenumber for different explicit and compact tridiagonal/pentadiagonal methods for interpolation and filtering.	26
2.11	A box function $\chi(x) = 1$, $x \in [4.5, 5.5]$ (red-solid) and a mollified box function $\bar{\chi}$ (green-dashed) with Eq. (2.51).	28

2.12	(a) Solution of the Poisson equation ($\nabla^2 u = f$) with the forcing term $f(x, y) = -2 \cos(x + y)$, $(x, y) \in \Omega = [0, 2\pi]^2$ and Dirichlet boundary conditions $u_b(x, y) = \cos(x + y)$, (x, y) on $\partial\Omega$ via an iterative fourth-order compact method. (b) Corresponding error contours $E(x, y) = u(x, y) - \cos(x + y) $, $(x, y) \in \Omega$ in comparison with the exact solution for $N = 1024^2$ grid points. (c) The 9-point stencil used in the fourth-order compact discretization of the Poisson equation on a two-dimensional grid. (d) Error analysis for Poisson solvers via the PSOR method using second and fourth order compact discretizations.	33
2.13	(top) The trigonometric basis functions for a complex FFT of a periodic function. (center) The trigonometric basis functions for a <i>sine</i> FFT of a function with homogeneous Dirichlet boundary conditions. (bottom) The trigonometric basis functions for a <i>cosine</i> FFT of a function with homogeneous Neumann boundary conditions. Picture from [56].	37
2.14	(a) Solution of the Poisson equation ($\nabla^2 u = f$) with a fourth-order compact method, forcing term is $f(x, y) = -(n^2 \pi^2 / L_y^2 + 1) \cos(x) \sin(n \pi y / L_y)$, $(x, y) \in \Omega = [0, 2\pi]^2$ and Dirichlet boundary conditions are given by $u_b(x, y) = \cos(x) \sin(n \pi y / L_y)$, (x, y) on $\partial\Omega$, ($n = 3$). (b) The corresponding error contours $E(x, y) = u(x, y) - u_{\text{exact}}(x, y) $, $(x, y) \in \Omega$ in comparison with the exact solution. (c) Error analysis for direct Poisson solvers computed with second and fourth order compact methods. (d) CPU-time scaling of different iterative (Multigrid / Point Successive Over Relaxation) and direct methods.	44
2.15	Stable regions for time integration via Adams–Bashforth and Runge–Kutta methods on complex plan. The picture is taken from [63].	45
2.16	Schematic representation of the fourth-order Runge–Kutta method. In each time step the <i>RHS</i> operator must be evaluated four times: once at the initial point (marked \bullet), twice at trial midpoints (marked \circ) and once at a trial endpoint (marked \circ). From these derivatives the value of the function in the next time step (marked \bullet) can be calculated. The picture is taken from [56].	45
2.17	Time accuracy of explicit fourth-order Runge–Kutta method applied to the Burgers equation over uniform grid with $\nu = 10^{-3}$	46
2.18	Flowchart of the fluid–solid interaction (FSI) algorithm.	46
3.1	Vorticity (a) and stream-function (b) contours for Taylor vortices, $(x, y) \in [0, 2\pi] \times [0, 2\pi]$ at $t = 0.0001$. Error analysis for vorticity (c) and stream-function (d), computed with second and fourth order finite differences. . .	50
3.2	Setup of a Taylor–Couette flow, picture from Wikipedia.	51
3.3	Schematic representation of a penalized unit square domain for modeling of Taylor–Couette flow with volume penalization method ($\chi = 0$ represents the fluid domain and $\chi = 1$ the solid domains respectively). The radius of the inner cylinder is $R_1 = 0.15$ and that of the outer cylinder is $R_2 = 0.4$. The angular velocity of the inner cylinder is $\Omega_1 = 0.2$ and that of the outer is equal to zero, $\nu = 0.01$ and $Ta \approx 1$	52

3.4	(a) Original and mollified mask function, comparison of computed vorticity ω with the exact solution, using $N = 128$ grid points in each direction. (b) Comparison of the computed stream-function ψ and velocity u with the exact solution, using $N = 128$. (c) The L_1 -error of u with the spatial resolution (N being the grid resolution in each direction). (d) The L_1 -error of u versus the penalization parameter η	54
3.5	Sketch of the domain $(\Delta X, \Delta Y) \in]1.2, 1.2[$ used to compute the hydrodynamic coefficients via control volume method (equivalent to control surface and surrounding lines in two dimensions).	57
3.6	Vorticity contours around a circular cylinder (simulation 4), where $Re=200$, $\Delta x = \frac{24}{4097}$, $\Delta y = \frac{12}{2049}$, $x_{cg} = L_x/4$, $y_{cg} = L_y/2$, $\Delta t = 10^{-3}$ and $\eta = 10^{-3}$. . .	60
3.7	Hydrodynamic forces and moment for the circular cylinder (simulation 3), where $Re=200$, $\Delta x = \frac{24}{1025}$, $\Delta y = \frac{12}{513}$, $\Delta t = 2 \times 10^{-3}$ and $\eta = 2 \times 10^{-3}$. . .	61
3.8	Vorticity contours of the flow around a circular cylinder (simulation 5) started by an initial perturbation ($u = U_\infty + 0.01 \times u'$, $v = 0.01 \times v'$) and noise in the inflow ($u = U_\infty + 10^{-4} \times u'$, $v = 10^{-4} \times v'$) where $u', v' \in [0, 1]$ are random numbers, $Re=200$, $\Delta x = \frac{24}{1025}$, $\Delta y = \frac{12}{513}$, $x_{cg} = L_x/4$, $y_{cg} = L_y/2$, $\Delta t = 10^{-3}$ and $\eta = 10^{-3}$	62
3.9	Pressure contours of the flow around a circular cylinder (simulation 6) started by an initial perturbation ($u = U_\infty + 10^{-3} \times u'$, $v = 10^{-3} \times v'$) and noise in inflow ($u = U_\infty + 10^{-4} \times u'$, $v = 10^{-4} \times v'$) where $u', v' \in [0, 1]$ are random numbers, $Re=200$, $\Delta x = \frac{24}{513}$, $\Delta y = \frac{12}{257}$, $x_{cg} = L_x/4$, $y_{cg} = L_y/2$, $\Delta t = 2 \times 10^{-3}$ and $\eta = 2 \times 10^{-3}$	63
3.10	(a) Comparison of the hydrodynamic forces of the circular cylinder, calculated via the surrounding control volume and the volume penalization method (simulation 6), started by an initial perturbation ($u = U_\infty + 10^{-3} \times u'$, $v = 10^{-3} \times v'$) and noise in inflow ($u = U_\infty + 10^{-4} \times u'$, $v = 10^{-4} \times v'$) where $u', v' \in [0, 1]$ are random numbers, $Re=200$, $\Delta x = \frac{24}{513}$, $\Delta y = \frac{12}{257}$, $\Delta t = 2 \times 10^{-3}$ and $\eta = 2 \times 10^{-3}$. (b) Components (pressure, momentum, volume and shear) of the drag force $Fx = -FxP - FxM - FxV + FxS$ calculated by the CV method. (c) Components (pressure, momentum, volume and shear) of the lift force $Fy = -FyP - FyM - FyV + FyS$ calculated by the CV method.	64
3.11	An example of the initial perturbation created with a hyperbolic tangent function for triggering the transition during the sedimentation of a cylinder $\mathbf{u}(\mathbf{x}) = f \times \text{noise} \times \mathbf{u}'$, where $f = 0.5(\tanh y' + 1)$, $\text{noise} = 10^{-3} \times U_{ref}$, $\mathbf{u}' \in [-1, 1]$ are uniformly distributed random numbers, $U_{ref} = \mathbf{u}_{max}$, $y' = 10y/L_y - 5$, $y' \in [-5, 5]$, $y \in [0, 1]$ and vorticity $\omega = v_x - u_y$	66
3.12	Vorticity isolines (dashed lines are used for negative values) of the falling cylinder in fully quiescent fluid, performed by the 2 nd -order solver, where free-slip boundary conditions are imposed at the surrounding walls, $g = -9.81 \text{ m/s}^2$, $\rho_b/\rho_f = 1.01$, $D = 0.005 \text{ m}$, $(x, y) \in [0, 0.04 \text{ m}] \times [0, 0.32 \text{ m}] = [0, 8 \text{ D}] \times [0, 64 \text{ D}]$, $\Delta t = 1.25 \times 10^{-4}$, the resolution is set to 512×4096 , the penalization parameter $\eta = 10^{-3}$, the filter parameter for denoising of the hydrodynamic coefficients is $\delta = 0.001$, $\nu = 8 \times 10^{-7} \text{ m}^2/\text{s}$ and $Re \approx 156$. .	67

3.13	Vorticity isolines (dashed lines are used for negative values) of the falling cylinder in a slightly perturbed fluid, performed by the 4 th -order solver, where free-slip boundary conditions are imposed at the surrounding walls, $g = -9.81 \text{ m/s}^2$, $\rho_b/\rho_f = 1.01$, $D = 0.005 \text{ m}$, $(x, y) \in [0, 0.04 \text{ m}] \times [0, 0.32 \text{ m}] = [0, 8D] \times [0, 64D]$, $\Delta t = 1.25 \times 10^{-4}$, resolution 4096×512 , penalization parameter $\eta = 5 \times 10^{-4}$, $\Delta t = 1.25 \times 10^{-4}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 8 \times 10^{-7} \text{ m}^2/\text{s}$ and $Re \approx 150$	68
3.14	Comparison of the streamwise $u_{\text{streamwise}}$ and lateral u_{lateral} velocities of the falling cylinder via different methods/parameters with reference simulations. Symbols indicate the simulations performed by Gazzola et al. [144] (red triangles) and Namkoong et al. [119] (green circles). Solid and dashed lines represent the results with the proposed algorithm on 4096×512 grid point with penalization parameter $\eta = 10^{-3}$, respectively performed by, the 4 th -order solver with a perturbed IC and free-slip BC (blue solid), the 2 nd -order solver with unperturbed IC and free-slip BC (black dashed), the 2 nd -order solver with unperturbed IC and no-slip BC (purple dash-dot) and the 2 nd -order solver with perturbed IC and free-slip BC (cyan dash-dot-dot) on the finest resolution 8192×1024 with penalization parameter $\eta = 10^{-4}$	69
3.15	Vorticity isolines (dashed lines are used for negative values) of the falling ellipse in the steady regime, where resolution of the grid is $Im \times Jm = 513 \times 2049$, $(x, y) \in [0, 5L] \times [0, 20L]$, $L = 2a = 1$, $J^* = 0.16$, $\rho_s/\rho_f = 1.538/1.0$, $g = -9.81$, $a/b = 0.5/0.1$, $X_0^{cg} = L_x/2$, $Y_0^{cg} = L_y - 3a$, $\theta_0 = \pi/4$, $\delta_{\text{filter}} = 10^{-3}$, $\eta = 10^{-3}$, $\nu = 0.03$ and $Re \approx 15$	70
3.16	Vorticity isolines (dashed lines are used for negative values) of the falling ellipse in the fluttering regime, where resolution of the grid is $Im \times Jm = 513 \times 2049$, $(x, y) \in [0, 5L] \times [0, 20L]$, $L = 2a = 1$, $J^* = 0.16$, $\rho_s/\rho_f = 1.538/1.0$, $g = -9.81$, $a/b = 0.5/0.1$, $X_0^{cg} = L_x/2$, $Y_0^{cg} = L_y - 3a$, $\theta_0 = \pi/4$, $\delta_{\text{filter}} = 10^{-3}$, $\eta = 10^{-3}$, $\nu = 0.01$ and $Re \approx 46$	70
3.17	Vorticity isolines (dashed lines are used for negative values) of the falling ellipse in the tumbling regime, where resolution of the grid is $Im \times Jm = 2049^2$, $(x, y) \in [0, 10] \times [0, 10]$, $J^* = 0.16$, $\rho_s/\rho_f = 1.538/1.0$, $g = -9.81$, $a/b = 0.5/0.1$, $X_0^{cg} = L_x/2$, $Y_0^{cg} = L_y - 3a$, $\theta_0 = \pi/4$, $\delta_{\text{filter}} = 10^{-3}$, $\eta = 10^{-3}$, $\nu = 0.003$ and $Re \approx 153$	71
3.18	Comparisons of (cg) trajectories of the falling ellipse, obtained in the present investigation, with those of Gazzola et al. [144].	72
3.19	Vorticity isolines (dashed lines are used for negative values) of the falling ellipse in different regimes, where $J^* = 0.16$, $\rho_b/\rho_f = 1.538$, $g = -9.81$, $a/b = 0.5/0.1$, $X_0^{cg} = L_x/2$, $Y_0^{cg} = L_y - 3a$, $\theta_0 = \pi/4$, $\delta_{\text{filter}} = 10^{-3}$ and $\eta = 10^{-3}$	74
3.20	(a) Hydrodynamic coefficients of a falling ellipse in the fluttering regime, where $J^* = 0.16$, $\rho_b/\rho_f = 1.538$, $a/b = 1/5$ and $\nu = 0.01$ before denoising. (b) After applying the first-order filter (2.25) with $b = 0$ and $\alpha = 0.2$. (c) After applying the second-order filter via Eqs. (2.25) and (2.26) with $\delta = 0.001$. (d) The corresponding velocity components.	75
4.1	Different types of fishes. Cartilaginous fishes (c-g), pictures are taken from [165].	78
4.2	Locomotion models and body types	79

4.3	Basic external anatomy of a lamprey, picture is taken from [165].	79
4.4	The haddock, a type of cod, is ray-finned. Pectoral fins (paired), ventral fins (paired), dorsal fin (three), adipose fin, anal fin (two) and caudal (tail) fin (one) are illustrated, picture is taken from [165].	80
4.5	Dimensions considered for a typical swimming fish: (a) side view and (b) top view, picture is taken from Eloy [157].	80
4.6	(a) Schematic view of the Bénard-von Kármán (BvK) vortex street behind a circular cylinder. (b) The reverse BvK (rBvK) vortex street in the backside of a swimming fish. (c) The average velocity difference $u(y)$ from the mean flow U in the far wake is a jet toward the cylinder. (d) In the case of swimming $u(y)$ is backward oriented in the center line. Both of these jets are surrounded by a region of counterflow. In an stable configuration of vortices each dipole creates a small jet represented by black vectors, pictures are taken from Eloy [157] with a slight modification.	82
4.7	Schematic three-dimensional views of the (a) BvK and (b) rBvK vortex streets, corresponding to the two-dimensional views of Fig. 4.6, picture is taken from Eloy [157].	83
4.8	Sketch of the two-dimensional Joukowski transform. (a) The original circle in the ζ plane where $r_c = 1$, $\eta_c = -0.1$ and $\theta_c = 0$. (b) The hydrofoil shape in the z plane. (c) The shape is rescaled to fit $0 \leq x \leq 1$	87
4.9	Backbone deformation according to Eq. (4.9) with $l = 1$ in one period, $\lambda = 0.5$, $a_0 = 0$, $a_1 = 0.01$ and $a_2 = 0.09$ to match the envelope used by Bergmann and Iollo [145].	88
4.10	Backbone deformation according to Eq. (4.9) with $l = 1$ in one period, $\lambda = 0.5$, $a_0 = 0.02$, $a_1 = -0.08$ and $a_2 = 0.16$ to match the experimental envelope measured by Videler and Hess [41].	88
4.11	Backbone deformation according to Eq. (4.9) with $l = 1$ in one period, $\lambda = -0.1$, $a_0 = 0.02$, $a_1 = -0.08$ and $a_2 = 0.16$ to match the experimental envelope measured by Videler and Hess [41].	89
4.12	Backbone deformation according to Eq. (4.9) with $l = 1$ in one period, $\lambda = -1.5$, $a_0 = 0.02$, $a_1 = -0.08$ and $a_2 = 0.16$ to match the experimental envelope measured by Videler and Hess [41].	90
4.13	Body fitted structured grid for covering the fish. The mesh is generated by the normal to the backbone lines.	91
4.14	Shape of the fish given by Eq. (4.6) before deformation.	96
4.15	Left: The Lagrangian structured grid ($Im_b \times Jm_b = 121 \times 19$) over the Eulerian one. Right: The Lagrangian structured grid covering the fish after deformation and the corresponding velocity vectors of each point, colored by absolute velocity $\sqrt{u^2 + v^2}$	96
4.16	Left: Interpolated mask function $\bar{\chi}$ from a Lagrangian grid ($Im_b \times Jm_b = 61 \times 9$) where $\Delta \mathbf{x} < \Delta \mathbf{X}$, as can be seen insufficient resolution in Lagrangian grid results in $\bar{\chi} = 1$ in very few points inside the fish. Right: Boundary of the Lagrangian grid (black line) and the interpolated smoothed mask function $\bar{\chi}$ (colored isolines). As can be seen the boundary of the Lagrangian grid ($Im_b \times Jm_b = 121 \times 19$) lies between minimum and maximum values of the mask function. The velocity components forming a jet at the tail while the fish is turning.	97

4.17	Smooth step function proposed by Boyer et al. [106] for gradually evolving the fish curvature in the first stroke $Cr(t) = t' - \sin(2\pi t')/(2\pi)$, $t \in [t_i, t_f]$ with $t' = (t - t_i)/(t_f - t_i)$, $t_i = 0$, $t_f = 1$. At $t = 1$ the left- and right-hand limits are equal for the function Cr and its first Cr' and second Cr'' derivatives.	100
4.18	Forward velocity U of a 2D anguilliform swimmer ($\lambda = f = 1$). Solid lines indicate the reference simulations performed by Kern and Koumoutsakos (green) [105] and Gazzola et al. (pink and brown) [144]. Dashed lines represent the results with the proposed algorithm.	101
4.19	Snapshots of vorticity isolines obtained during a simulation in a rectangular domain $(x, y) \in [0, 10l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, with resolution 2048×1024 , by imposing a penalization parameter inside the body equal to $\eta = 10^{-3}$ and the time step $\Delta t = 10^{-3}$. The filter parameter for denoising of the hydrodynamic forces is $\delta_{\text{filter}} = 0.005$ and the Reynolds number is approximately $Re \approx 3800$	105
4.20	Schematic representation of the desired angle for curvature control, during the rotation of the fish toward the goal. Here $\theta_{\text{des}} = \theta_{\text{goal}} - \theta_{\text{Head}}$ is the difference of the angles between head direction and the line passing through the target and the head ($-\pi < \theta_{\text{des}} < \pi$), picture adapted from Bergmann and Iollo [145] with a slight modification.	106
4.21	Snapshots of vorticity isolines obtained during a simulation of swimming fish toward a predefined target which is located at $(x_f, y_f) = (0.9L_x, 0.5L_y)$. At $t = 0$ the fish and the surrounding flow are at rest. After reaching the vicinity ($r = 0.5l_{\text{fish}}$) of the target the curvature of the backbone tends to zero by Eq. (4.26). The domain of the solution is $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, the resolution of the Eulerian grid is 1024×1024 , the resolution of the Lagrangian grid ($Im_b \times Jm_b = 251 \times 39$), the penalization parameter $\eta = 5 \times 10^{-4}$ and the kinematic viscosity is equal to $\nu = 1.4 \times 10^{-4}$. Samples of the backbone of the fish are plotted in Fig. 4.25.	107
4.22	Snapshots of pressure isolines obtained during a simulation of swimming fish (represented by black contour corresponding to $\chi = 0.2$) toward a predefined goal which is located at $(x_f, y_f) = (0.9L_x, 0.5L_y)$. At $t = 0$ the fish and the surrounding flow are at rest. After reaching the vicinity ($r = 0.5l_{\text{fish}}$) of the target the curvature of the backbone tends to zero by Eq. (4.26). The domain of the solution is $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, the resolution of the Eulerian grid is 1024×1024 , the resolution of the Lagrangian grid ($Im_b \times Jm_b = 251 \times 39$), the penalization parameter $\eta = 5 \times 10^{-4}$ and the kinematic viscosity is equal to $\nu = 1.4 \times 10^{-4}$	108
4.23	Saddle points are denoted by green dashed circles and vortices are denoted by purple solid circles. The vortices forming dipoles during the rotation, corresponding to strong jets.	109
4.24	Stream-function ψ isolines during the rotation of the fish. Saddle points are denoted by green dashed circles and vortices are denoted by blue (counter-clockwise) and red (clockwise) solid circles.	110

4.25	Samples of the backbone of a swimming fish toward a predefined goal which is located at $(x_f, y_f) = (0.9L_x, 0.5L_y)$ obtained during a simulation, $t \in [0, 15]$. After reaching the vicinity ($r = 0.5l_{\text{fish}}$) of the goal the curvature of the backbone tends to zero by Eq. (4.26). The snapshots of the corresponding vorticity and pressure isolines are plotted in Figs. 4.21 and 4.22. Starting from rest the fish performs a 180° rotation within an area of about 1.3 times its length.	111
5.1	Sparse point representation of 1D functions, obtained by WT with cubic interpolation ($J = 10$), filtered with threshold $\epsilon = 1 \times 10^{-3}$. The green dots (marked ●) show the retained grid points. Left: Gaussian function, compression = 95%, L_∞ -Error $\leq 1 \times 10^{-4}$. Right: Function (5.8), Compression = 94%, L_∞ -Error $\leq 5 \times 10^{-5}$	118
5.2	(a) Vorticity contours of dipole-wall collision at $t = 0.4$. (b)-(d) Comparison of the total energy $E(t)$, the total enstrophy $Z(t)$ and the total palinstrophy $P(t)$ between the data from Clercx and Bruneau [104] and the present finite-difference computation with a uniform multigrid solver for Reynolds 1000 with different grid resolutions. By increasing the resolution the results converge toward the reference simulation.	121
5.3	The evolution and collision of the vortices (represented by colored isolines) with walls (left) and the corresponding adaptive grid (right), the maximum grid level is $J = 11$ in each direction, threshold $\epsilon = 10^{-3}$, and Reynolds 1000.	125
5.4	Comparisons of the total energy (a) and the total enstrophy (b) between the uniform grid solver and the multiresolution computation with thresholds, $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$, for Reynolds number 1000 and a maximum grid level $J = 9$ in each direction for all simulations. (c) Convergence study for the total enstrophy $Z(t)$ toward the data from Clercx and Bruneau [104] with the uniform grid solver for 256^2 , 512^2 and 1024^2 grid points and multiresolution computations with a maximum grid level $J = 11$ in each direction, for Reynolds 1000. It can be observed that coarse grid computations are too dissipative, we anticipate that 4098^2 grid resolution is sufficient to reproduce the results of Clercx and Bruneau [104].	126
5.5	Up: Forward velocity U of the anguilliform 2D swimmer ($\lambda = f = 1$). Solid lines indicate the reference simulations performed by Kern and Koumoutsakos (green) [105], Gazzola et al. (pink and brown) [144] and Ghaffari et al. (red and blue) [170]. Dashed lines represent the results with the proposed multiresolution algorithm. Down: Evolution of the number of active grid points, significant pints (corresponding to the retained points after filtering of wavelet coefficients), safety zone, hung and interpolated points for the wavelet transform during the computation with the multiresolution solver, with a maximum grid level $J = 10$ in each direction (1025^2 grid points). . .	127
5.6	Adaptive grids colored by the vorticity (left) and colored by the mask $\bar{\chi}$ (right) at $t = 6$ (zoom in, from up to down) where $(x, y) \in [0, 8l_{\text{fish}}] \times [0, 8l_{\text{fish}}]$ by imposing the penalization parameter inside the body equal to $\eta = 10^{-3}$, with maximum grid level of $J = 10$ in each direction and the viscosity $\nu = 1.4 \times 10^{-4}$	128

5.7	Adaptive grids at different instances colored by the vorticity of a swimming fish where $(x, y) \in [0, 8l_{\text{fish}}] \times [0, 8l_{\text{fish}}]$. The penalization parameter inside the body equals $\eta = 10^{-3}$, with a maximum grid level of $J = 10$ in each direction and the viscosity is $\nu = 1.4 \times 10^{-4}$	129
6.1	(a) Arrangement of velocity \bullet and pressure \circ grids (shifted in x and y directions by $\Delta x/2$ and $\Delta y/2$) in a two-dimensional half-staggered arrangement, physical boundaries are represented by black lines (—), ghost nodes for pressure are on the blue lines ($\circ - \circ - \circ$), (b) Indices of velocity \bullet and pressure \bullet nodes in a two-dimensional half-staggered grid.	136
6.2	An example of two dimensional domain decomposition using 4×3 (row \times column) processors. For data in the X-pencils one global operation in the Z-pencils direction needs 4 data transpositions to come back to the X-pencils, i.e., $X \rightarrow Y \rightarrow Z$ (operation) $\rightarrow Y \rightarrow X$. Pencil rotation (transpose) is done via the MPI “ALL TO ALL” subroutine.	149
6.3	Some characteristics of the Incompact3d code in terms of scaled wavenumber in comparison to other methods, pictures are taken from Lamballais et al. [150].	150
6.4	The Q isosurfaces of the falling sphere in a quiescent fluid, obtained with the penalized Incompact3d solver, where $g = -20$, $\rho_b/\rho_f = 1.041$, $D = 1$, $(x, y, z) \in [0, 12] \times [0, 4] \times [0, 4]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-2}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 10^{-2}$ and $Re \approx 100$	151
6.5	The z -mid section velocities of the falling sphere in a quiescent fluid, obtained with the penalized Incompact3d solver, where $g = -20$, $\rho_b/\rho_f = 1.041$, $D = 1$, $(x, y, z) \in [0, 12] \times [0, 4] \times [0, 4]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-2}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 10^{-2}$ and $Re \approx 100$	152
6.6	The z -mid section velocities (up) and streamlines colored by streamwise velocity (down) of the falling sphere in a quiescent fluid obtained with the penalized Incompact3d solver, where $g = -20$, $\rho_b/\rho_f = 1.041$, $D = 1$, $(x, y, z) \in [0, 12] \times [0, 6] \times [0, 6]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 10^{-3}$ using AB2, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-2}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 10^{-2}$ and $Re \approx 100$	153
6.7	The streamlines colored by streamwise velocity of the falling sphere in a quiescent fluid at $t = 12$, obtained with the penalized Incompact3d solver, where $g = -20$, $\rho_b/\rho_f = 1.041$, $D = 1$, $(x, y, z) \in [0, 12] \times [0, 6] \times [0, 6]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 10^{-3}$ using AB2, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-2}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 10^{-2}$ and $Re \approx 100$	154
6.8	Comparison of the streamwise velocity V_x of the falling sphere ($g = 20$) using different penalization parameters η , with the reference simulation performed by Kern and Koumoutsakos [105] (black line). Colored lines represent the results of the present study performed with the penalized Incompact3d solver, with $\Delta t = 10^{-4}$ using AB3.	155
6.9	The corresponding streamwise force and the displacement of the falling sphere, represented in Fig. 6.8, by imposing the penalization parameter to $\eta = 10^{-2}$	155

6.10	Backbone of the fish as a one-dimensional Cosserat medium. The inertial frame is denoted by (x, y, z) , the orthogonal body fitted coordinate (d_1, d_2, d_3) oriented along the body to be tangent in d_3 direction. Picture taken from Lazarus et al. [154].	156
6.11	Profiles of the considered fish from top and side.	156
6.12	The fish is constructed by series of ellipses normal to the backbone of the considered fish. Each ellipse is covered by a structured grid.	157
6.13	The surface of the considered fish is covered by a Lagrangian structured grid.	157
6.14	The corresponding velocities, evaluated by Eq. (6.52), of the swimming fish at the surface of the Lagrangian structured grid.	158
6.15	The interpolated mask function χ and the velocity components, on the Eulerian grid.	158
6.16	The mask function with two different resolutions of the Lagrangian grid.	159
6.17	Q iso-surfaces of the swimming fish obtained with the penalized Incompact3d solver, where $l = \lambda = f = 1$, $\rho_b = \rho_f$, $(x, y, z) \in [0, 3] \times [0, 1] \times [0, 1]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 2 \times 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-3}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 2 \times 10^{-3}$ and $Re \approx 100$	161
6.18	The streamlines colored by streamwise velocity of the swimming fish obtained with the penalized Incompact3d solver, where $l = \lambda = f = 1$, $\rho_b = \rho_f$, $(x, y, z) \in [0, 3] \times [0, 1] \times [0, 1]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 2 \times 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-3}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 2 \times 10^{-3}$ and $Re \approx 100$	162
6.19	The z -mid velocity field of the swimming fish obtained with the penalized Incompact3d solver, where $l = \lambda = f = 1$, $\rho_b = \rho_f$, $(x, y, z) \in [0, 3] \times [0, 1] \times [0, 1]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 2 \times 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-3}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 2 \times 10^{-3}$ and $Re \approx 100$	163
6.20	The forces (top-left), velocities (top-right) and the trajectories (bottom) of the swimming fish obtained with the penalized Incompact3d solver, where $l = \lambda = f = 1$, $\rho_b = \rho_f$, $(x, y, z) \in [0, 3] \times [0, 1] \times [0, 1]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 2 \times 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-3}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 2 \times 10^{-3}$ and $Re \approx 100$. The reference point is the head.	164
8.1	Domaine de la solution et du corps immergé, $\Omega = \Omega_f \cup \Omega_p$	170
8.2	Une représentation schématique du domaine de la solution pour le solveur rapide de l'équation de Poisson. Les opérations dans les directions x et y sont découplées. Dans la direction x des conditions aux limites générales peuvent être utilisées grâce aux schémas aux différences finies. Dans la direction y des conditions aux limites d'imperméabilité et de glissement (Dirichlet homogène, i.e., $\psi = \omega = 0$) sont imposées permettant d'utiliser la transformée en <i>sinus</i>	174
8.3	Profil du poisson donné par l'équation (8.24) avant déformation.	176
8.4	Modèle de déformation de la colonne vertébrale selon l'équation (8.25) pendant une période avec $a_0 = 0.02$, $a_1 = -0.08$, $a_2 = 0.16$, $L = 1$ et $\lambda = -1.5$	177

8.5	(a) Étapes de constructions du maillage structuré avec les lignes normales à la colonne vertébrale sur chaque point discret. (b) Maillages lagrangiens structurés (mobiles et déformables) qui recouvrent le poisson nageant. . . .	178
8.6	(a) Maillage lagrangien structuré couvrant le poisson en déformation et les vitesses correspondantes de chaque point, colorées suivant l'intensité de la vitesse (absolue) $\sqrt{u^2 + v^2}$. (b) Maillage lagrangien structuré composé de $Im_b \times Jm_b = 121 \times 19$ points.	180
8.7	Organigramme de l'algorithme d'interaction fluide-structure.	181
8.8	(a) Fonction proposée par Boyer et al. (2006) [106] pour accroître progressivement la courbure de la colonne vertébrale du poisson : $Cr(t) = t' - \sin(2\pi t')/(2\pi)$, $t \in [t_i, t_f]$ avec $t' = (t - t_i)/(t_f - t_i)$, $t_i = 0$ et $t_f = 1$. A $t = 0$ et $t = 1$ les limites à gauche et les limites à droite sont égales pour la fonction Cr et pour ses dérivées première Cr' et seconde Cr'' . (b) Fonction proposée pour estimer la courbure désirée $k_{des}(\theta)$ suivant l'équation (8.38) avec $k_{max} = \pi$ et $\theta_{limit} = \pi/4$	185
8.9	Une représentation schématique de l'angle désiré pour contrôler la courbure rajoutée k_{offset} (le long de la colonne vertébrale) du poisson afin de le diriger vers son objectif. Ici $\theta_{des} = \theta_{objectif} - \theta_{tete}$ est l'angle entre la direction de la tête et la ligne reliant la tête à la position de son objectif, ($-\pi < \theta_{des} < \pi$). Image adoptée d'après Bergmann et Iollo (2011) [145] avec quelques modifications.	186
8.10	Champs de vorticit� (a-f) et de pression (g-l) autour du poisson (repr�sent� par les lignes noires correspondant � $\chi = 0.2$) nageant vers un objectif pr�d�fini, situ� au point $(x_f, y_f) = (0.9L_x, 0.5L_y)$. A $t = 0$, le poisson et le fluide environnant sont au repos. La domaine de la solution est $(x, y) \in [0, 5l_{fish}] \times [0, 5l_{fish}]$, la r�solution du maillage eul�rien est 1024×1024 , la r�solution du maillage lagrangien est $(Im_b \times Jm_b = 251 \times 39)$, le param�tre de p�nalisation est $\eta = 5 \times 10^{-4}$ et la viscosit� cin�matique est �gale � $\nu = 1.4 \times 10^{-4}$	188
8.11	Les points selles (entour�s des cercles en pointill�s verts) et centres (entour�s des cercles solides violets) dans l'�coulement s�par� autour du poisson sont successivement lâch�s par le mouvement du corps. Deux tourbillons forment un dip�le qui g�n�re un jet localis� vers l'arri�re dans l'�coulement au cours de la nage du poisson.	189
8.12	Les positions successives du poisson mat�rialis�es par sa colonne vert�brale au cours de sa nage vers l'objectif pr�d�fini situ� � $(x_f, y_f) = (0.9L_x, 0.5L_y)$ sont repr�sent�es pour un intervalle de temps $t \in [0, 15]$. A proximit� de l'objectif ($r_{objectif} = 0.5l_{poisson}$) la courbure de la colonne vert�brale du poisson, donn�e par l'�quation (8.37), se ram�ne � z�ro. Les champs de vorticit� et de pression correspondants sont illustr�s sur la Figure 8.10. Le poisson initialement au repos effectue un changement de direction de 180° pr�s du bord gauche du domaine dans une aire qui correspond � environ 1.3 fois sa longueur.	189

D.1	(top) The trigonometric basis functions for a <i>complex</i> FFT of a periodic function. (center) The trigonometric basis functions for a <i>sine</i> FFT of a function with homogeneous Dirichlet boundary conditions. (bottom) The trigonometric basis functions for a <i>cosine</i> FFT of a function with homogeneous Neumann boundary conditions. Picture from [56].	197
-----	---	-----

List of Tables

2.1	The considered convergence criteria for the residual $ Res $ of the Poisson solvers, for the problem ($\nabla^2 u = f$) presented in Fig. 2.12, using different resolutions.	32
2.2	The number of iterations of the Poisson ($\nabla^2 u = f$) solvers, for the problem presented in Fig. 2.12, using different resolutions.	34
2.3	CPU-times (second / 100 CALL) for different (multigrid, point successive over relaxation and direct) Poisson solvers, for the problem ($\nabla^2 u = f$) presented in Fig. 2.14, using different resolutions. The processor is Intel(R) Core(TM) i5-3230M CPU@2.6 GHz.	38
2.4	Maximum values of modified wavenumbers for the first and second derivatives via central finite difference methods.	41
2.5	Stability limits of some explicit methods via linear analysis [63].	41
3.1	Boundary conditions for ψ and ω on a rectangular domain $\Omega \in [0, L_x] \times [0, L_y]$ which is used for the simulation of the flow around circular cylinder.	55
3.2	Comparison of the parameters/methods used for simulation of the unsteady flow around circular cylinder at $Re = 200$	56
3.3	Comparison of the results obtained from the developed code for simulation of the unsteady flow around a circular cylinder at $Re = 200$ with those of other researchers. Comparison is done for Strouhal number (determined from the time variation of lift), hydrodynamic coefficients (lift, drag and moment) and the transition time over the curve of the lift coefficient.	58
6.1	Coefficients of the third-order Runge–Kutta method.	140
6.2	The possible boundary conditions in the Incompact3d code, (ℓ is power of 2, 3, 4, 5 and 6).	142
A.1	Coefficients of explicit differentiation ($\alpha = \beta = 0$) for the first f' and the second derivative f'' with Eqs. (A.1) and (A.6).	192
A.2	Coefficients of implicit compact differentiation (via tri-diagonal system of equations, $\beta = 0$) for the first derivative f' with Eq. (A.1).	192
A.3	Coefficients of implicit compact differentiation (via tri-diagonal system of equations, $\beta = 0$) for the second derivative f'' with Eq. (A.6).	192
A.4	Coefficients of spectral-like (formally fourth-order) implicit compact differentiation (via five-diagonal system of equations) for the first derivative f' with Eq. (A.1) for inner points.	192

A.5	Coefficients of implicit compact differentiation (which leads to a pentadiagonal system of linear equations) for the near boundary points for evaluation of the first derivative f' via Eq. (A.11). Optimised by Kim [111] to keep the fourth-order accuracy near the boundaries.	193
-----	---	-----

Chapter 1

Introduction

Moriarty: How are you at Mathematics?

Seegoon: I speak it like a native.

Spike Milligan (The Goon Show)

During the last decades, great advances have been achieved in numerical simulation of fluid flows. New mathematical ideas, algorithms, models, mesh generation techniques, linear system solvers, parallel processing, etc. has been developed rapidly. At the same time revolution in hardware capacities, helped the researchers to go far from the imagination of the CFD pioneers in terms of computation power and storage capacity. Therefore, flow solvers have become versatile, robust and accurate thanks to the large amount of research projects. Nowadays, the maturity of numerical algorithms makes possible the integration and coupling of new physical phenomena to deal with more realistic problems such as: high Reynolds turbulent flows, multi-phase flows, etc. Simulation of fluid–structure interaction is one of these interdisciplinary fields of interest as is explained by Leroyer and Visonneau [100] (2005). The quantification and simulation of the flow around biological swimmers is another challenge in fluid mechanics (Sotiropoulos and Yang [167], 2014). At the same time bio-inspired design of swimming robots are in growth (El Rafei et al. [120], 2008). The costs of experimental studies (Belkhiri [158], 2013) lead the researchers to develop efficient predictive numerical algorithms for hydrodynamic analyses of fish swimming. Difficulties of numerical simulations of fish-like swimming are due to different reasons. One problem is efficient quantification of the kinematics of different species (more than 32,000) which seems to be far from the simple laws proposed in different studies. Efficient simulation of incompressible flows is also an important problem, where the efficiency of the elliptic solver is crucial. The third bottleneck in numerical simulations of swimming is the coupling of the fluid solver with deformable, moving and rotating bodies. Fishes swim by exerting force and torque against the surrounding water. This is normally done by the fish contracting muscles on either side of its body in order to generate moving waves from head to tail.

These waves generally are getting larger as they go toward the tail (Wikipedia contributors [166], 2014). The resultant force exerted on the water by such motion generates a force (even oscillatory) which pushes the fish forward. Most fishes generate thrust moving their body and fins. In general these movements can be divided into *undulatory* and *oscillatory* motions. Mechanisms of locomotion using body and fins are divided into groups that differ in the fraction of their body that is displaced laterally (Breder [6], 1926). *Anguilliform* swimmers are long and slender, in which there is little increase in the amplitude of the flexion wave as it passes along the body. In *carangiform* swimmers, there is a more remarkable increase in wave amplitude along the body with the vast majority of the work being done by the rear half of the fish. In *thunniform* fishes almost all of the lateral movement takes place at the tail. *Ostraciiform* fishes have no appreciable body wave when they employ caudal locomotion, only the tail fin itself oscillates rapidly to create thrust. However there are other minorities (Wikipedia contributors [166], 2014). The tail beat creates a reversed Kármán street of vortices and generates thrust, leaving thus a momentumless wake back. By varying the frequency and amplitude of the oscillation a variety of wakes, like classical Kármán, two pairs (2P) (Van Rees et al. [160], 2013), two pairs plus two single (2P+2S), etc. can be observed (Williamson and Roshko [65], 1988; Schnipper et al. [128], 2009).

Anguilliform fishes add a constant curvature to their backbone for turning, i.e., they use their body like a rudder for torque generation. Yeo et al. [134] (2010) studied numerically the straight swimming/cruising and sharp turning manoeuvres in two-dimensions. They have shown that a carangiform-like swimmer execute a sharp turn through an angle of 70° from straight coasting within a space of about one body length. Gazzola et al. [155] (2012) investigated the C-start escape patterns of a larval fish by using a remeshed vortex particle approach and the volume penalization method. To maximize the escape distance, the deformation of the fish based on the mid-line curvature values, is optimized via an evolutionary strategy, developed by Hansen et al. [90] (2003). Bergmann and Iollo [145] (2011) performed numerical simulations of fish rotation and swimming toward a prescribed goal. They considered the average profile of the fish backbone aligns over a circle with an estimated radius to perform a rotation. The radius of the circle tends to infinity ($r \rightarrow \infty$) in a forward gait. The considered fish by Bergmann and Iollo [145] (2011) is constructed by a complex valued mapping like the Kutta–Joukowski transform superposed to the fish backbone with prescribed undulatory motion. Here we will present a simple law for rotation control of an anguilliform fish. Our rotation control law (Bontoux et al. [168], 2014) is similar to that presented by Yeo et al. [134] (2010), and Bergmann and Iollo [145] (2011), in which the feedback is based on the angle between the line-of-sight and the direction of surge. But instead of adding a radius to the backbone, we envisage to use curvature which seems to be more efficient. We use the method proposed by Boyer et al. [106] (2006) which is based on quaternions for efficient description of the fish backbone kinematics. We apply the rotation control to two-dimensional swimming. Even if due to the shape and

deformation style of the fish-like swimmers the surrounding flow is fully three dimensional, most of the fundamental features of swimming are included in two-dimensional analyses.

For simulation of incompressible flows the Navier–Stokes equations can be reformulated in vorticity-velocity (Gazzola et al. [144], 2011) or vorticity stream-function (Spotz and Carey [61], 1995) formulations. For two-dimensional problems the vorticity formulation is reduced to a scalar valued evolution equation. Hence only the vorticity transport equation has to be advanced in time. The choice of finite differences in the present investigation is related to the use of an immersed boundary method where a Cartesian grid can be used. Therefore the use of finite differences is efficient and straightforward. Among finite difference methods high-order compact discretizations, (Hirsh [33], 1975; Lele [55], 1992), are more advantageous in terms of accuracy and reasonable cost. We refer to Abide and Viazzo [97] (2005) and Boersma [146] (2011) for high-order compact discretizations of the incompressible Navier–Stokes equations in primitive variables and to Bontoux et al. [35] (1978), Roux et al. [36] (1980), and Spotz and Carey [61] (1995) for compact high-order solutions of the vorticity and stream-function formulation. Solving the incompressible Navier–Stokes equations typically implies an elliptic Poisson equation which is the most time consuming part of the algorithm. Direct methods like diagonalization or FFT based solvers can be used. Iterative methods, namely, point successive over relaxation (PSOR) with red-black sweeper, multigrid or Krylov subspace solvers are other alternatives. Using high-order discretizations iterative methods are less attractive because the resulted matrices are less sparse, thus the rates of convergence are slow. However iterative methods can cover all types of boundary conditions, we refer to Spotz and Carey [61] (1995) for a fourth-order compact discretization of the Poisson equation. On the other hand, in direct methods the memory limitation is restrictive for simulations on a fine grid. Therefore decoupling of the directions by FFT based methods can be advantageous, even if it implies some limitations in the boundary conditions. We propose a direct fourth-order solver for the Poisson equation which is a combination of a compact finite difference with a sine FFT in alternative directions. The main advantages of our method are fourth-order accuracy, efficiency, the possibility to parallelize and convergence down to zero machine precision over an optimal grid. Other advantages and limitations of the proposed solver are discussed in the Chapter 2.

A difficulty in numerical simulations of fish swimming is the analysis of fluid/solid interaction, which can be handled by strong or loose coupling in accordance with implicit or explicit time advancement, we refer to Sotiropoulos and Yang [167] (2014) for a detailed discussion. We use the volume penalization method, known also as Darcy-Brinkmann penalization (Brinkmann [9], 1947), proposed by Arquis and Caltagirone [40] (1984), Angot et al. [73] (1999) and Khadra et al. [75] (2000). This method belongs to the diffuse-interface immersed boundary methods (IBMs). It consists of modeling the immersed body as a porous medium, thus getting rid of the Dirichlet boundary conditions by considering both

the fluid and the body as one domain with different permeabilities. So one can consider a rectangular solution domain in which the body is immersed and can even move. The penalization method leads to between first and second order accuracy near the body and is an efficient method in dealing with deformable, moving and rotating bodies immersed in a fluid. A development to deal with rigid bodies colliding with each other in incompressible flows is performed by Coquerelle and Cottet [123] (2008). An extension to include elasticity of the solid interacting with fluid via the volume penalization method is done by Engels et al. [159] (2013). We refer to the review of Mittal and Iaccarino [99] (2005) for a complete classification and description of immersed boundary methods. One advantage of this class of penalization schemes for fluid–structure interaction problems is that it enables the use of time and space adaptivity via multiresolution analysis, as recently demonstrated by Gazzola et al. [162] (2014) and Ghaffari et al. (2014). An extension of the two-dimensional solver to be space adaptive, based on multiresolution analysis, including the penalization term, is done in Chapter 5. In the introduction of Chapter 5 the state of the art and remaining open problems of multiresolution analysis for incompressible flows is discussed.

In this thesis, we focus on some numerical aspects of efficient turning laws for anguilliform swimmers, a topic which is less studied so far. To this end the geometrically exact theory of nonlinear beams based on quaternions (developed by Boyer et al. [106], 2006) is adapted to the backbone kinematics description. Starting by the code developed by Sabetghadam et al. [127] (2009) we apply compact finite differences to the vorticity stream-function formulation of the Navier–Stokes equations including the penalization term. An efficient direct method is presented for solving the Poisson equation. Then different numerical aspects of the algorithm like accuracy in space and the error introduced by the penalization method are examined. In Chapter 6 an extension to perform a three dimensional simulation of swimming fish is presented. The Incompact3d open access code developed by Laizet and Lamballais [129] (2009) is used. The code is modified to be able to take into account the deformable bodies interaction with incompressible flows via the implicit volume penalization method. The report is organized as follows; First in Chapter 2 our methodology including the governing equations, discretization and the algorithm for fluid interaction with solid bodies in two-dimensions is presented. Next validation of the algorithm is carried out via different test cases and different errors are studied in Chapter 3. Then the algorithm for fluid interaction with forced deformable bodies in two-dimensions and kinematics of an anguilliform swimmer is presented in Chapter 4. After validation of the proposed algorithm, the results for swimming and rotation control are reported. An extension to perform a multiresolution analysis of swimming fish is done in Chapter 5, by applying the Harten’s point value data representation to the developed finite difference solver. Further extension to three-dimensional simulation of a swimming fish is done in Chapter 6. Finally, the results are discussed and some guides for future works are addressed in Chapter 7. The extended summary of thesis in French is given in Chapter 8.

Chapter 2

An algorithm for fluid–structure interaction in two dimensions

“Numerical simulation is half-way between theory and experiment without replacing either, since theory, simulation and experiment are all interrelated.”

Marie Farge [118] (2007)

The third way to study nature

This Chapter presents some fundamental concepts of *discretization of the incompressible Navier–Stokes equations in the presence of complex geometries*, which will be used in the present investigation. First the governing equations of incompressible flows, i.e., the Navier–Stokes equations are recalled and then reformulated in vorticity and stream-function form. Then the boundary conditions are reviewed in summary and the *volume penalization method* is presented for simulation of flow around complex geometries. The volume penalization method provide the hydrodynamic forces and torques in an straightforward manner, however to cope with the fluid–structure interaction (FSI) problems denoising of the hydrodynamic coefficients is used. By using an immersed boundary method a Cartesian grid can be used, therefore a high order finite-difference discretization is motivated. *Compact methods* provide a general frame for construction of different implicit and explicit formula for high-order differentiation, filtering and interpolation. Then finite-difference discretizations in the compact form with different accuracies are examined on uniform and stretched grids. The Poisson equation, which has an elliptic nature, is regularly encountered in the simulations of incompressible flows to insure the mass conservation. Therefore the accuracy and efficiency of the elliptic solver has crucial importance in the performance of the algorithm. To this end a new direct fourth-order solver for the Poisson equation is proposed to ensure the incompressibility constraint down to machine zero on an opti-

mal grid. Moreover a review to explicit time integration methods is done and finally an algorithm for numerical simulation of fluid–structure interaction is proposed.

2.1 Vorticity-stream function formulation

The governing equations of incompressible flows are the Navier–Stokes equations (2.1) - (2.2). Using primitive variables, the momentum equation reads

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho_f} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F} \quad (2.1)$$

and the (mass conservation) continuity equation corresponds to a divergence-free velocity field,

$$\nabla \cdot \mathbf{u} = 0 \quad , \quad \mathbf{x} \in \Omega \in \mathbb{R}^2 \quad (2.2)$$

where Ω is the spatial domain of interest, given as an open subset of \mathbb{R}^2 , which can be bounded or unbounded in general, $\mathbf{u}(\mathbf{x}, t)$ is the velocity field, $p(\mathbf{x}, t)$ is the pressure, $\nu = \mu/\rho_f > 0$ is the kinematic viscosity of the fluid, ρ_f is the density of the fluid and $\mathbf{F}(\mathbf{x}, t)$ is a source term. The Navier–Stokes equations are written for unit mass of the fluid, therefore the dimension of the terms are like acceleration, i.e., $[LT^{-2}]$. For a complete description of a particular problem, the above equations need to be complemented to describe an initial/boundary value problem (IBVP). Hence by specifying an initial condition for the velocity field

$$\mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0(\mathbf{x}) \quad \text{with} \quad \nabla \cdot \mathbf{u}_0 = 0$$

which we assume to be in C^∞ and divergence free in all of Ω , and by giving boundary conditions for velocity

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_{BC}(\mathbf{x}, t) \quad , \quad \mathbf{x} \in \partial\Omega$$

one will seek the solution during time evolution. Following McDonough [114] to guarantee existence of a solution to a given problem it is required that the boundary conditions satisfy a *consistency* or *compatibility* condition of the form

$$\int_{\partial\Omega} \mathbf{u}_{BC} \cdot \mathbf{n} \, dA = 0$$

which express the global mass conservation and \mathbf{n} is an outward pointing vector normal to the boundary $\partial\Omega$. The consistency condition is obvious for the case of $\mathbf{u}_{BC} = 0$ which represents the no-slip and no-penetration (solid wall) boundary condition. The pressure is well defined and unique up to an additive constant. Although, in general, the pressure is time dependent only its gradient appears in the Navier–Stokes equations (2.1). Thus, in the procedure of solution pressure does not need an initial condition. By considering

a homogeneous Neumann boundary conditions for the pressure at inflow, free-slip and no-slip walls

$$\frac{\partial p}{\partial \mathbf{n}} = 0 \quad \text{on } \partial\Omega \quad (2.3)$$

the Navier–Stokes equations are completed [103]. At the outflow the value of p_∞ can be fixed while the velocity components can be extrapolated. Nevertheless at the inflow the velocity components are fixed and the pressure can be extrapolated using Eq. (2.3). With the use of homogeneous Neumann boundary conditions at all boundaries, the value of the pressure at one point in the solution domain must be fixed, to guarantee the existence and convergence of the solution. By choosing respectively L , ρ_f and U_∞ as reference length, density and velocity for a given problem

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{L}, \quad \tilde{\mathbf{u}} = \frac{\mathbf{u}}{U_\infty}, \quad \tilde{t} = \frac{t}{L/U_\infty}, \quad \tilde{p} = \frac{p}{\rho_f U_\infty^2}, \quad \tilde{\mathbf{F}} = \frac{\mathbf{F}}{U_\infty^2/L}$$

the Navier–Stokes equations can be written in non-dimensional form in which $Re = U_\infty L/\nu$ is the Reynolds number

$$\frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{t}} + (\tilde{\mathbf{u}} \cdot \tilde{\nabla}) \tilde{\mathbf{u}} = -\tilde{\nabla} \tilde{p} + \frac{1}{Re} \tilde{\nabla}^2 \tilde{\mathbf{u}} + \tilde{\mathbf{F}} \quad (2.4)$$

However, in two-dimensional problems the vorticity and stream-function formulation has the advantage that it not only eliminates the pressure variable entirely, but also ensures a divergence free velocity field (mass conservation), if Eq. (2.6) is properly satisfied. One encounters two scalar quantities, i.e., ψ and ω , instead of the velocity vector and the pressure field, thus it makes the computations more efficient. We retain this formulation in the following, but the presented concepts are applicable also to the primitive variable formulation. By taking the curl of the momentum Eq. (2.1), after elimination of the vortex stretching term due to the two-dimensional assumption and elimination of the baroclinic term due to the constant density, one obtains the vorticity transport equation

$$\partial_t \omega + (\mathbf{u} \cdot \nabla) \omega = \nu \nabla^2 \omega + \nabla \times \mathbf{F}, \quad \mathbf{x} \in \Omega \in \mathbb{R}^2 \quad (2.5)$$

for two-dimensional flows, where $\omega(\mathbf{x}, t) = \nabla \times \mathbf{u} = v_x - u_y$ denotes the vorticity. The vorticity transport equation (2.5) is a parabolic equation and the velocity components are determined from $\mathbf{u} = -\nabla^\perp \psi$, i.e., $(u, v) = (\partial_y \psi, -\partial_x \psi)$ with ψ being the stream function, satisfying

$$-\nabla^2 \psi = \omega \quad (2.6)$$

which is an elliptic¹ equation for ψ . By considering proper boundary conditions Eq. (2.6) may be solved numerically via FFT-based direct methods, iterative methods like successive over relaxation or multigrid methods. The efficiency of the method used to solve the elliptic part of the problem is crucial for the accuracy and efficiency of the whole algorithm. Advantages and limitations of different elliptic solvers will be discussed later. With the use of auxiliary relations for the velocity components it is possible to eliminate the velocity vector from Eq. (2.5), to obtain:

$$\frac{\partial \omega}{\partial t} = -\frac{\partial \psi}{\partial y} \frac{\partial \omega}{\partial x} + \frac{\partial \psi}{\partial x} \frac{\partial \omega}{\partial y} + \nu \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) \quad (2.7)$$

An equation for pressure can be derived by applying the divergence operator to momentum equation and making use of the continuity,

$$\nabla \cdot (\nabla p) = -\rho_f \nabla \cdot [(\mathbf{u} \cdot \nabla) \mathbf{u}] + \rho_f \nabla \cdot \mathbf{F} \quad (2.8)$$

Without the forcing term in two-dimensions, we thus have

$$\nabla^2 p = 2\rho_f (u_x v_y - u_y v_x) \quad (2.9)$$

the right hand side of Eq. (2.9) can also be deduced from the stream-function, i.e., we have $\nabla^2 p = 2\rho(\psi_{xx}\psi_{yy} - \psi_{xy}^2)$. Proper boundary conditions for pressure must be considered.

Boundary conditions

The boundary conditions for a curved boundary (s) moving tangentially to its surface with a constant velocity U_{tan} can be written in terms of the stream-function ψ , at each boundary section Γ_i . The no-penetration boundary condition is equivalent to

$$\left. \frac{\partial \psi}{\partial \hat{\tau}} \right|_{wall} = \mathbf{n} \cdot \mathbf{u}(s, t) = 0 \quad (\text{Neumann}) \quad \Leftrightarrow \quad \psi|_{wall} = C_i \quad (\text{Dirichlet}) \quad (2.10)$$

the free-slip boundary condition on the surface can be imposed easily by

$$\omega = 0 \quad (\text{Dirichlet})$$

and the no-slip boundary condition reads,

$$\left. \frac{\partial \psi}{\partial \mathbf{n}} \right|_{wall} = -\hat{\tau} \cdot \mathbf{u}(s, t) = \pm U_{tan} \quad (\text{Neumann}) \quad (2.11)$$

¹Perturbations will spread in all directions with the speed of sound which approaches to infinity in the incompressible limit.

where C_i is a constant for each Γ_i , (s) is the representing curve of the surface, $\hat{\tau}$ is the direction tangent to the wall and \mathbf{n} is the direction normal to the wall. For a wall with zero tangential velocity we have $U_{tan} = 0$. In a simply-connected domain, C_0 can be taken equal to zero ($C_0 = 0$). As a result of the above formulation for a fixed horizontal/vertical wall the no-penetration and no-slip boundary conditions are ($v = u = 0$) or ($\psi_x = \psi_y = 0$). Inflow/outflow boundary conditions can be defined by $\psi_{\hat{\tau}} = U_{\infty}$, $\psi_{\mathbf{n}} = 0$, respectively. In two dimensional incompressible flows, beside the mentioned advantages of vorticity and stream-function formulation, one can mention some disadvantages. Yet, the main difficulties in the numerical implementation of this formulation come from the boundary conditions [84]. The majors among them are as follows:

1. The implementation of the two cited boundary conditions for the stream-function simultaneously.
2. When the vorticity must be updated in time, there is no definite boundary condition for vorticity.
3. Determining the constants C_i at each boundary of 'holes' Γ_i if the computational domain is multi-connected.

However, there are several methods to update the vorticity boundary condition over a solid boundary moving tangent to its surface with a constant velocity U_{tan} . In the following we cite some formulas for the vorticity at the wall, which will lead to second and fourth order accuracy:

Thom's [8] formula

$$\omega_{i,0} = \frac{\psi_{i,0} - \psi_{i,1} \pm hU_{tan}}{0.5h^2} + O(h^2)$$

Jensen's [16], Wilkes (1963), Pearson (1965) and Roache [30] formula

$$\omega_{i,0} = \frac{7\psi_{i,0} - 8\psi_{i,1} + \psi_{i,2} \pm 6hU_{tan}}{2h^2} + O(h^2)$$

Briley's [29] formula

$$\omega_{i,0} = \frac{85\psi_{i,0} - 108\psi_{i,1} + 27\psi_{i,2} - 4\psi_{i,3} \pm 66hU_{tan}}{18h^2} + O(h^4)$$

Other relations were proposed by Woods [11], Orszag and Israeli [32]. It is very important to know that, the vorticity boundary condition is responsible to enforce no-slip boundary condition. Although at the boundaries, none of these relations can force the velocity components down to machine zero, however the accuracy of the normal to the wall component of the velocity is two orders of magnitude larger than the tangent component. The subject of the vorticity boundary condition has a long history, going back to Thom's formula in

[8]. In a second-order scheme, Thom's formula, Wilkes formula, or some other local formulas can be selected and coupled with a centered finite difference scheme at the interior points. The advantage of Thom's formula lies in its simplicity and stability as only one interior point of the stream-function is involved. Yet, it was always very confusing why formulas like Thom's, which seem hopelessly to be first-order by formal Taylor expansion on the boundary, are actually second-order accurate. This mystery can be explained by Strang-type high order expansions [57]. A review of vorticity boundary conditions can be found in [62], [52] and [72]. In Gresho [52] the advantage of integral approach for boundary conditions over differential approach is demonstrated.

2.1.1 Volume penalization method

For the simulation of flow in the presence of curved solid boundaries which do usually not coincide with grid points, one can use the volume penalization method which is proposed by Arquis and Caltagirone [40] and Angot et al. [73], [75]. It is based on the idea of modeling solid bodies as porous media, thus getting rid of the Dirichlet boundary conditions by considering both the fluid and the solid part as one domain with different permeabilities, so one has a domain in which the solid is embedded. This method will lead to between first and second order accuracy near the solid boundaries, that will be demonstrated for the Taylor–Couette flow in Section 3.2. In the Navier–Stokes equations (2.1) in primitive variables or the vorticity transport equation (2.5), the penalization term can be added as a forcing term (or damping term) and thus, it is possible to introduce a solid body into the flow field. The penalization term for unit mass of the fluid reads,

$$\mathbf{F} = -\eta^{-1}\chi(\mathbf{u} - \mathbf{u}_B) \quad (2.12)$$

where $\mathbf{u}_B(\mathbf{x}, t)$ is the velocity vector of the immersed body which will be zero for fixed bodies. The penalization parameter η is the permeability coefficient of the immersed body with dimension $[T]$. For an explicit time integration of the governing equations Δt must be smaller than η to ensure the stability of time integration. Typically values of permeability ranging from $\eta = 10^{-4}$ up to 10^{-2} . The mask (characteristic) function χ is dimensionless and describes the geometry of the immersed body, see Fig. 2.1

$$\chi(\mathbf{x}, t) = \begin{cases} 1 & \mathbf{x} \in \Omega_p \\ 0 & \mathbf{x} \in \Omega_f \end{cases} \quad (2.13)$$

where Ω_f represents the domain of the flow and Ω_p represents the immersed body in the solution domain. The solution domain is governed by the Navier–Stokes equations in the fluid regions and by the Darcy–Brinkmann law in the penalized regions, in the limit when $\eta \rightarrow 0$. The volume penalization method is also subjected to a stiffness problem associated

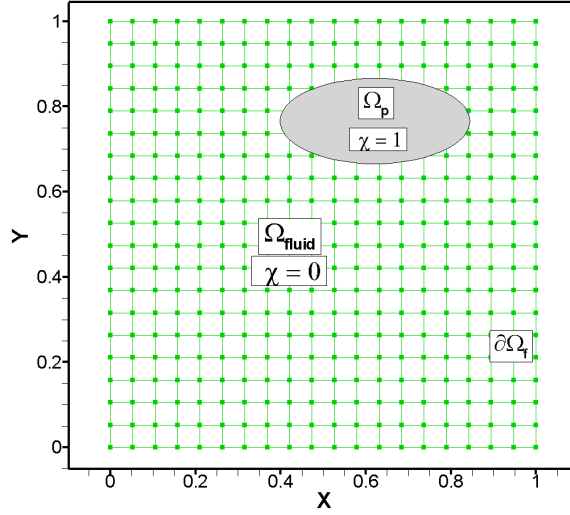


Figure 2.1: Domain of the solution and the immersed body, $\Omega = \Omega_f \cup \Omega_p$.

with large variation in the values of $\eta^{-1}\chi$ [99]. Therefore for stabilizing the numerical solution and also for accuracy enhancement, filtering of the mask function χ has been proposed by researchers. Following Forestier [79] and Minguez [122] the mask function is mollified by the Shuman [15] filter

$$\bar{\chi}_{i,j} = (2\chi_{i,j} + \chi_{i+1,j} + \chi_{i-1,j} + \chi_{i,j+1} + \chi_{i,j-1})/6 \quad (2.14)$$

which is equivalent to a raised *cosine* filter in Fourier space, we refer to Pasquetti et al. [121] for more details. The following definition, which is proposed by Gazzola et al. [144] for the mask function χ , is an alternative, where the mollification of the mask function is included in the definition

$$\chi(\mathbf{x}, t) = \begin{cases} 0 & d < -\epsilon \\ 0.5(1 + d/\epsilon + \frac{1}{\pi} \sin(\pi d/\epsilon)) & |d| \leq \epsilon \\ 1 & d > \epsilon \end{cases}$$

where d is the signed distance from the surface of the body (negative outside, positive inside) and ϵ is the mollification length. As a general rule, for moderate Reynolds numbers ($Re < 10000$), ϵ should be a small fraction ($< 1\%$) of the characteristic length of the geometry under study, e.g., $\epsilon = 2.8h$, h being the grid size. Consequently, it defines the curvature of the finest resolved features of the object [144]. It should also allow the mollified characteristic function to span 4-5 grid points in order to have a numerically stable and accurate normal [144] to the immersed boundary.

If u_η^{num} denotes the numerical solution of the penalized equations, for quantifying the nu-

merical error of u_η^{num} compared to the solution of the original Navier–Stokes problem u^{exact} , the error can be estimated by

$$\|u^{\text{exact}} - u_\eta^{\text{num}}\| \leq \underbrace{\|u^{\text{exact}} - u_\eta\|}_{O(\sqrt{\eta})} + \underbrace{\|u_\eta - u_\eta^{\text{num}}\|}_{O(\Delta x^p)} \quad (2.15)$$

where $\|\cdot\|$ is an appropriate norm. The first term at the right-hand side is the error due to the penalization term and the second term represents the discretization error (p being the formal order of accuracy of the numerical method used to discretize the penalized equation). A compromise between these two errors is to choose $\Delta x \approx \sqrt{\eta}$, which will lead to a first-order bound for the error

$$\|u^{\text{exact}} - u_\eta^{\text{num}}\| \leq O(\Delta x) \quad (2.16)$$

As mentioned in Section 2.1 in the vorticity-stream function formulation the curl of the force term must be added to the right hand side of the Eq. (2.5). This can be written as

$$\nabla \times \mathbf{F} = \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} = \eta^{-1} \left(\frac{\partial}{\partial y} (\chi(u - u_B)) - \frac{\partial}{\partial x} (\chi(v - v_B)) \right)$$

or

$$\nabla \times \mathbf{F} = \eta^{-1} \left(\frac{\partial}{\partial y} (\chi \frac{\partial \psi}{\partial y}) + \frac{\partial}{\partial x} (\chi \frac{\partial \psi}{\partial x}) - \frac{\partial}{\partial y} (\chi u_B) + \frac{\partial}{\partial x} (\chi v_B) \right)$$

For points in which the complete stencil belongs to the fluid domain ($\chi = 0$) we have $\nabla \times \mathbf{F} = 0$. While for points in which the complete stencil belongs to the solid domain ($\chi = 1$) we have $\nabla \times \mathbf{F} = -\eta^{-1}(\omega - \omega_B)$, i.e., in the time integration, we have

$$\omega^{n+1} \approx \omega_B = 2\Omega_B + \omega_{\text{def}}$$

where Ω_B is the angular velocity of the embedded body and ω_{def} is the vorticity due to the deformation of the immersed body. The penalization term is thus responsible for the vorticity production at the walls where the stencil of discretization belongs to both solid and fluid domains. The volume penalization term can also be written in the following form:

$$\nabla \times \mathbf{F} = \underbrace{-\eta^{-1}\chi(\omega - \omega_B)}_{\text{Volume penalization}} + \underbrace{\eta^{-1}\chi_y(u - u_B) - \eta^{-1}\chi_x(v - v_B)}_{\text{Vorticity production at the surfaces}} \quad (2.17)$$

Theoretically derivatives of the discontinuous mask function (2.13) corresponds to Dirac delta function (i.e., at a discontinuity $\chi_x \rightarrow \infty$ or $\chi_y \rightarrow \infty$). However, in numerical evaluations near the discontinuities the derivatives of the the mask function have limited values. Thus the spatial derivatives of the mask function, i.e., χ_x and χ_y can be approximated numerically or analytically. An analytical relation for the smooth Dirac delta function is

proposed by Lai and Peskin [76]

$$\delta(r) = \begin{cases} \frac{1}{8}(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}) & |r| \leq 1 \\ \frac{1}{8}(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}) & 1 < |r| \leq 2 \\ 0 & |r| > 2 \end{cases} \quad (2.18)$$

where $r = (\mathbf{x} - \mathbf{X}_B)/\Delta\mathbf{x}$. Or simply the following relation

$$\delta(r) = \begin{cases} \frac{1}{4}(1 + \cos(\pi|r|/2)) & |r| \leq 2 \\ 0 & |r| > 2 \end{cases} \quad (2.19)$$

can be used for estimation of the derivatives of characteristic function. As can be seen in Fig. 2.2 the derivative of the filtered (smoothed) mask function $\bar{\chi}_x$ evaluated by classical fourth-order Padé scheme, coincides exactly with the analytical functions proposed by Lai and Peskin [76] in the context of forcing term evaluation for a diffused-interface immersed boundary method. An over prediction in the value of derivative of the smoothed mask function $\bar{\chi}_x$ via second-order FDM in comparison to compact fourth-order Padé scheme can be noticed in Fig. 2.2 (b).

Although there is some criticism to put the volume penalization method in the family of immersed boundary methods which impose a force at the near boundary nodes. But in the review paper of Mittal and Iaccarino [99] the sentence: “*In this review, we use the term immersed boundary (IB) method to encompass all such methods that simulate viscous flows with immersed (or embedded) boundaries on grids that do not conform to the shape of these boundaries.*” in the introduction of the paper can help to clarify why the volume penalization method can be classified as IBM. Moreover the volume penalization method, introduced by Angot et al. [73] and Khadra et al. [75], which also called Brinkmann [9] penalization, is cited in the review paper of Mittal and Iaccarino [99]. Our argument can be completed with the following explanation. Even if the force is applied to the entire immersed body, the main contributions are around the immersed boundaries. Considering the forcing term in Eq. (2.17) added to the vorticity equation, the vorticity production at the solid surfaces is evident. Moreover, numerical evaluation of the derivatives of the mask function χ_x and χ_y is resulted in a smooth Dirac δ function, similar to that proposed by Lai and Peskin [76], see again Fig. 2.2 (b). Finally, we conclude that the volume penalization method is a diffuse-interface IBM in contrast to what is named sharp-interface IBM in [99]. An equation for the pressure can be derived by applying the divergence operator to the momentum equation (with $\rho = \text{cte}$) and making use of the continuity,

$$\nabla \cdot (\nabla p) = -\rho_f \nabla \cdot [(\mathbf{u} \cdot \nabla) \mathbf{u}] - \frac{\rho_f}{\eta} \nabla \cdot [\chi(\mathbf{u} - \mathbf{u}_p)] \quad (2.20)$$

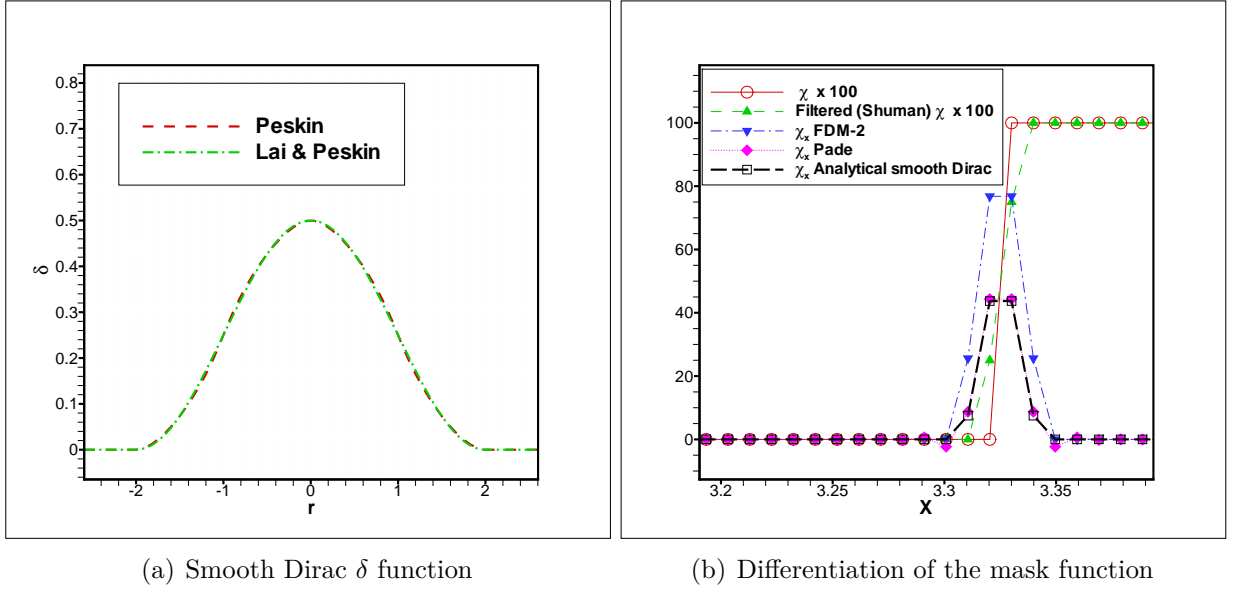


Figure 2.2: (a) Smooth Dirac δ function (2.18) proposed by Lai and Peskin [76]. (b) Comparison of the derivatives of the smoothed mask function computed via the second-order central finite difference method and the fourth-order compact finite difference (Padé) method with the analytical smoothed Dirac delta function (2.18).

2.1.2 Evaluation of the hydrodynamic coefficients

With the use of the volume penalization method, the hydrodynamic forces and the torques acting on the body, which are usually evaluated via surface integrals of the stress tensor $\sigma(\mathbf{u}, p) = \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - p\mathbf{I}$, can be computed readily by integrating the penalized velocity over the volume of the body (surface in two-dimensions), thus we have the force vector expressed in Newton [126]

$$\mathbf{F}^* = \oint_{\partial\Omega_s} \sigma \cdot \mathbf{n} \, dl = \lim_{\eta \rightarrow 0} \frac{\rho_f}{\eta} \int_{\Omega_s} \chi(\mathbf{u} - \mathbf{u}_B) \, ds + \rho_f S_{pen} \ddot{\mathbf{X}}_{ref} \quad (2.21)$$

for unit mass ($m = \rho_f v$) of the fluid we have $\mathbf{F} = \mathbf{F}^*/m$

$$\mathbf{F} \approx \frac{1}{\eta S_{pen}} \int_{\Omega_s} \chi(\mathbf{u} - \mathbf{u}_B) \, ds + \ddot{\mathbf{X}}_{ref} \quad (2.22)$$

in three-dimensions S_{pen} must be replaced by V_{pen} which is the volume of the immersed body. The torque $[N.m]$ in two-dimensions is evaluated by

$$M_{ref} = \oint_{\partial\Omega_s} \mathbf{r} \times \sigma \cdot \mathbf{n} \, dl = \lim_{\eta \rightarrow 0} \frac{\rho_f}{\eta} \int_{\Omega_s} \chi \mathbf{r} \times (\mathbf{u} - \mathbf{u}_B) \, ds + \frac{\rho_f}{\rho_s} I_{zz} \ddot{\theta}_{ref} \quad (2.23)$$

where $\mathbf{r} = (x - X_{ref})^2 + (y - Y_{ref})^2$, $I_{zz} = \int \mathbf{r}^2 dm$ is the moment of inertia taken around the reference point which can be the center of the gravity (cg), \mathbf{n} is the outward unit vector normal to $\partial\Omega_s$, θ is the angle of rotation with respect to the reference point, dots denote

derivation with respect to time and S_{pen} is the surface of the penalized area.

In dealing with fluid/solid interaction problems, the oscillation of the hydrodynamic forces and moments during successive iterations calculated from Eqs. (2.22) and (2.23) may cause some trouble in correctly predicting the accelerations. By moving the body the oscillations of the hydrodynamic coefficients is even more, we refer to Luo et al. [153] for more discussion. The difficulty comes from the fact that the hydrodynamic forces and torques acting on the body are used in the calculation of the linear and angular accelerations which in turn has an impact on the predicted velocity and trajectory of the solid. The oscillations are due to the nature of the penalization method, insufficient resolution, the approximative nature of Eqs. (2.22) and (2.23). The oscillations are like a noise and lead to invalid results and may even to the divergence of the simulations. An efficient method to eliminate them is to apply a low-pass filter, like *exponential smoothing*, usually used in denoising of data from time series

$$\hat{F}^n = \alpha F^n + (1 - \alpha) \hat{F}^{n-1} \quad , \quad \alpha \in [0, 1] \quad , \quad n = 2, 3, \dots \quad (2.24)$$

where $\hat{F}^1 = F^1$. The filter was first suggested by Brown in 1956 [13]. Then it is used by Kern et al. [105] to denoise the hydrodynamic forces and torques with $\alpha = 0.2$. Simple exponential smoothing is not efficient when there is a trend in the data. In such situations, several methods were devised like *second-order (double) exponential smoothing*

$$\hat{F}^n = \alpha F^n + (1 - \alpha)(\hat{F}^{n-1} + b^{n-1}) \quad , \quad n = 3, 4, \dots \quad (2.25)$$

$$b^n = \beta(\hat{F}^n - \hat{F}^{n-1}) + (1 - \beta)b^{n-1} \quad , \quad (\alpha, \beta) \in [0, 1] \quad (2.26)$$

where $\hat{F}^1 = F^1$, for $n = 2$ one can use Eqs (2.25) and (2.26) with $\alpha = \beta = 1$. Then $\alpha = 1 - (1 - \delta)^2$ and $\beta = \delta^2/\alpha$ can be used in which δ is a small band. According to our experiments, we propose $\delta \in [10^{-2}, 10^{-4}]$ for denoising of the hydrodynamic forces and torques. However, smaller values, e.g., $\delta = 5 \times 10^{-4}$, have a strong damping effect, larger values, e.g., $\delta = 5 \times 10^{-3}$, have less damping effect but there is a risk of divergence in the simulations. A sensitivity analysis must be done for each test case, see also the discussion of the results in Section 3.4.2 and Section 4.3.

2.1.3 Body dynamics

The dynamics of an arbitrary solid or deformable body moving in a viscous incompressible fluid is governed by Newton's second law

$$\Sigma(\mathbf{F}_H + \mathbf{F}_G) = m\ddot{\mathbf{X}}_{ref} \quad (2.27)$$

where the applied forces have been decomposed into two components; the hydrodynamic forces \mathbf{F}_H and the forces due to gravity $\mathbf{F}_G = V_b(\rho_b - \rho_f)\mathbf{g}$. Newton's law can be integrated directly to give the position of the reference point (which can be the center of mass) as a function of time. Holding \mathbf{F} constant over the discrete physical time step (Δt) gives

$$\Delta \mathbf{X}_{ref} = \frac{1}{2} \frac{\mathbf{F}^n}{m} \Delta t^2 + \mathbf{V}^n \Delta t \quad (2.28)$$

and $V^{n+1} = V^n + \dot{X} \Delta t$. The rotational motion is described by Euler's equation

$$\Sigma M = \frac{d}{dt}(I_{zz} \dot{\theta}) \quad (2.29)$$

where M is the applied torque around the reference point and $I_{zz} = J_z$ is the moment of inertia around the reference point which is equivalent to polar moment of inertia around the axis passing through the reference point $J_z = I_x + I_y$. Time integration of Eq. (2.29) regardless of changes in moment of inertia ($I_{zz} = cte$) and M , yields the new angle of the body with respect to a given reference

$$\Delta \theta = \frac{1}{2} \ddot{\theta}^n \Delta t^2 + \dot{\theta}^n \Delta t \quad (2.30)$$

where $\ddot{\theta} = M/I_{zz}$ and $\dot{\theta}^{n+1} = \dot{\theta}^n + \ddot{\theta} \Delta t$ (the dots represent derivation with respect to time). Eqs (2.28) and (2.30) describing a motion with three degrees of freedom (3DOF) for the considered body. In these equations second-order terms can be eliminated as done by Gazzola et al. [144] but we retain these terms. Eqs. (2.22) and (2.23) provide the fluid forces and torques necessary to integrate the system of ODEs formed by Eqs. (2.27) and (2.29). Denoising of hydrodynamic coefficients is done by Eq. (2.25). Appropriate initial conditions are necessary. In the present computations, we use a first-order scheme for time integration of the dynamic equations which seems to be sufficient because of the errors introduced by the penalization method which is also near first-order. The same time integration method is used by Kolomenskiy and Schneider [126] and Gazzola et al. [144] for the dynamics of the body where the penalization is also used.

2.2 Compact methods

In this section, the evaluation of spatial derivatives (discretization) on uniform and non-uniform grids, filtering and interpolation via compact methods [55] are presented and the introduced errors are evaluated via analytical expressions or numerical tests. Classical finite differences are based on Lagrange interpolation. Therefore high-order approximations lead to large stencils. In compact finite differences Hermit interpolation is used to keep high accuracy and compact stencil.

2.2.1 Spatial discretization on uniform grids

Given the values of a function f on a uniformly spaced mesh $x_i = (i - 1)h$, ($i = 1, \dots, N$) where $h = L_x/(N - 1)$, the derivatives of the function can be evaluated with different order of accuracy using the relations introduced by Lele [55]. Different explicit or implicit schemes for the approximation of the first derivative can be constructed from the following general relation

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = a \frac{f_{i+1} - f_{i-1}}{2h} + b \frac{f_{i+2} - f_{i-2}}{4h} + c \frac{f_{i+3} - f_{i-3}}{6h} \quad (2.31)$$

a similar relation for approximation of the second derivative is

$$\beta f''_{i-2} + \alpha f''_{i-1} + f''_i + \alpha f''_{i+1} + \beta f''_{i+2} = a \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + b \frac{f_{i+2} - 2f_i + f_{i-2}}{4h^2} + c \frac{f_{i+3} - 2f_i + f_{i-3}}{9h^2} \quad (2.32)$$

We refer to Appendix A for the coefficients of some prevalent methods with different orders of accuracy and their near boundary treatments. Choosing $\alpha = \beta = 0$ in Eqs. (2.31) and (2.32) result in point-wise explicit methods. If $\beta \neq 0$ and $\alpha \neq 0$ a linear system of equations with pentadiagonal coefficient matrix is obtained. With $\beta = 0$ a linear system of equations like (2.33) with tridiagonal coefficient matrix can be constructed. Linear system of equations with tri or penta diagonal coefficient matrix can be solved efficiently by direct lower-upper decomposition methods, like the Thomas algorithm (see Appendix F).

$$\begin{bmatrix} 1 & \beta & 0 & \dots & \dots & 0 & 0 & 0 & 0 \\ \gamma & 1 & \gamma & 0 & \dots & \dots & 0 & 0 & 0 \\ 0 & \alpha & 1 & \alpha & 0 & \dots & \dots & 0 & 0 \\ \vdots & 0 & \alpha & 1 & \alpha & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & \alpha & 1 & \alpha & 0 & \vdots \\ 0 & 0 & \dots & \dots & 0 & \alpha & 1 & \alpha & 0 \\ 0 & 0 & 0 & \dots & \dots & 0 & \gamma & 1 & \gamma \\ 0 & 0 & 0 & 0 & \dots & \dots & 0 & \beta & 1 \end{bmatrix} \begin{bmatrix} f'_1 \\ f'_2 \\ f'_3 \\ \vdots \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} RHS_1 \\ RHS_2 \\ RHS_3 \\ \vdots \\ \vdots \\ \vdots \\ RHS_{N-2} \\ RHS_{N-1} \\ RHS_N \end{bmatrix} \quad (2.33)$$

Compact methods can also be cast in general matrix forms:

$$[P]\mathbf{f}' = \mathbf{R} \quad , \quad [Q]\mathbf{f}'' = \mathbf{V} \quad (2.34)$$

where $[P]$ and $[Q]$ represent $N \times N$ diagonal matrices. Two direct solvers for tridiagonal and pentadiagonal linear systems of equations are described in Appendix F. As an alternative

$[P]$ and $[Q]$ can be inverted and stored once in a preprocessing step. In that case we have

$$\mathbf{f}' = [P]^{-1}\mathbf{R} \quad , \quad \mathbf{f}'' = [Q]^{-1}\mathbf{V} \quad (2.35)$$

where $[P]^{-1}$ and $[Q]^{-1}$ are full matrices and no more diagonal (band limited). In the later case the matrix multiplication can be vectorized over vector processors.

2.2.2 Spatial discretization on non-uniform grids

On non-uniform grids the derivatives can be evaluated with the use of the chain rule and mapping relations. For first derivatives we have

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} \quad , \quad \frac{\partial}{\partial y} = \eta_y \frac{\partial}{\partial \eta}$$

and for second derivatives

$$\frac{\partial^2}{\partial x^2} = \xi_x^2 \frac{\partial^2}{\partial \xi^2} + \xi_{xx} \frac{\partial}{\partial \xi} \quad (2.36)$$

$$\frac{\partial^2}{\partial y^2} = \eta_y^2 \frac{\partial^2}{\partial \eta^2} + \eta_{yy} \frac{\partial}{\partial \eta} \quad (2.37)$$

where ξ_x , ξ_{xx} , η_x and η_{yy} are the metrics of the transformation. For a uniform grid with $\Delta x = \Delta \xi$ and $\Delta y = \Delta \eta$ we have; $\xi_x = \eta_y = 1$ and $\xi_{xx} = \eta_{yy} = 0$. The distribution of grid points with stretching near boundaries, $x \in [0, L_x]$, is given by the following relation, see Fig. 2.3 (a),

$$x(\xi) = \frac{L_x}{2} \left\{ 1 - \frac{\tanh[\gamma(\beta - \xi)]}{\tanh(\gamma\beta)} \right\} \quad (2.38)$$

where L_x is the size of the domain in the x -direction and ξ represents the coordinate of the points which are uniformly distributed in the computational domain, $\xi \in [0, (Imax - 1)\Delta\xi]$ and β is the position of the inflection point. Therefore by choosing $\beta = \xi_{max}/2$ a symmetric grid will be obtained, where ξ_{max} is the length of the computational domain. The slope of the function at the inflection point, and thus the rate of stretching is determined by γ . Typically γ can be chosen equal to $\gamma = 4$. A very small number leads to a uniform grid. The metrics are given analytically

$$\xi_x = \frac{1}{x_\xi} = \frac{\tanh(\gamma\beta)}{\gamma L_x/2} \left\{ \cosh[\gamma(\beta - \xi)] \right\}^2$$

by using Eq. (2.36)

$$\frac{\partial^2 x}{\partial x^2} = \xi_x^2 \frac{\partial^2 x}{\partial \xi^2} + \xi_{xx} \frac{\partial x}{\partial \xi} \Rightarrow \xi_{xx} = -\xi_x^2 \frac{x_{\xi\xi}}{x_\xi} = -\xi_x^3 x_{\xi\xi}$$

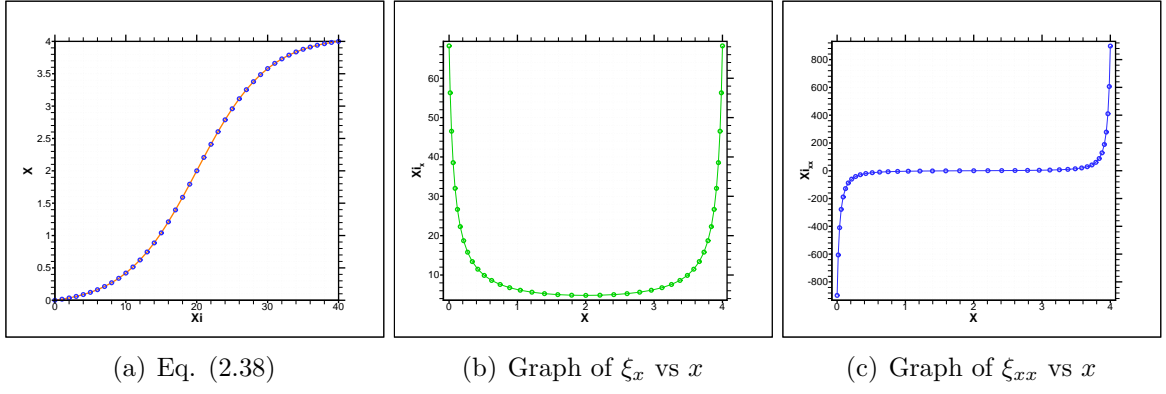


Figure 2.3: One-dimensional stretched grid, $x \in [0, L_x]$, with clustering near boundaries via Eq. (2.38), with $N = 41$, $L_x = 4$, $\gamma_x = 4$ and $\Delta\xi = 1$.

$$\xi_{xx} = -2\gamma \left[\frac{\tanh(\gamma\beta)}{\gamma L_x/2} \right]^2 \cosh^3[\gamma(\beta - \xi)] \sinh[\gamma(\beta - \xi)]$$

See Fig. 2.3 for a stretched grid and the corresponding metrics. For two given functions a truncation error analyses of the first and second derivatives using fourth-order differentiation on uniform and clustered grids are plotted in Fig. 2.4.

2.2.3 Analysis of differentiation errors

To assess the introduced errors in the procedure of differentiation, a function $f(x) = \sin x$, $x \in [0, 2\pi]$, on a uniform grid is considered. The results of the truncation error analysis (in physical space) for the first and second derivatives via different explicit and compact methods on uniform grids are plotted in Fig. 2.5. In comparison to high-order differencing schemes, low-order schemes with larger numerical error would require higher resolution to achieve the same degree of accuracy. In computational fluid dynamics spatial discretization errors have two primary (coupled) components: differentiation error and aliasing error associated with the nonlinear terms. Fourier analysis, and the concept of the *modified wavenumber* is useful in quantifying the differentiation error in dealing with high wavenumber oscillations. Error analyses in physical space do not have the possibility to show the ability of the differentiation method to deal with high wavenumber oscillations, which usually are present in the pressure field associated with acoustic waves or the velocity field affected by turbulence. The ability of the differentiation method to deal with high wavenumber oscillations, i.e., the scaled modes between $w \in [\pi/2, \pi]$, is affected by the accuracy of the numerical method which can be analyzed in Fourier space (via wavenumbers) over all of the possible modes $w \in [0, \pi]$. The results of error analyses by Fourier modes are rather independent of the formal order of accuracy of the method obtained via Taylor expansion. As shown in the following a formal fourth-order method proposed by Kim [111] can perform better than a formal eight-order method (via Taylor expansion) in dealing with high wavenumber oscillations.

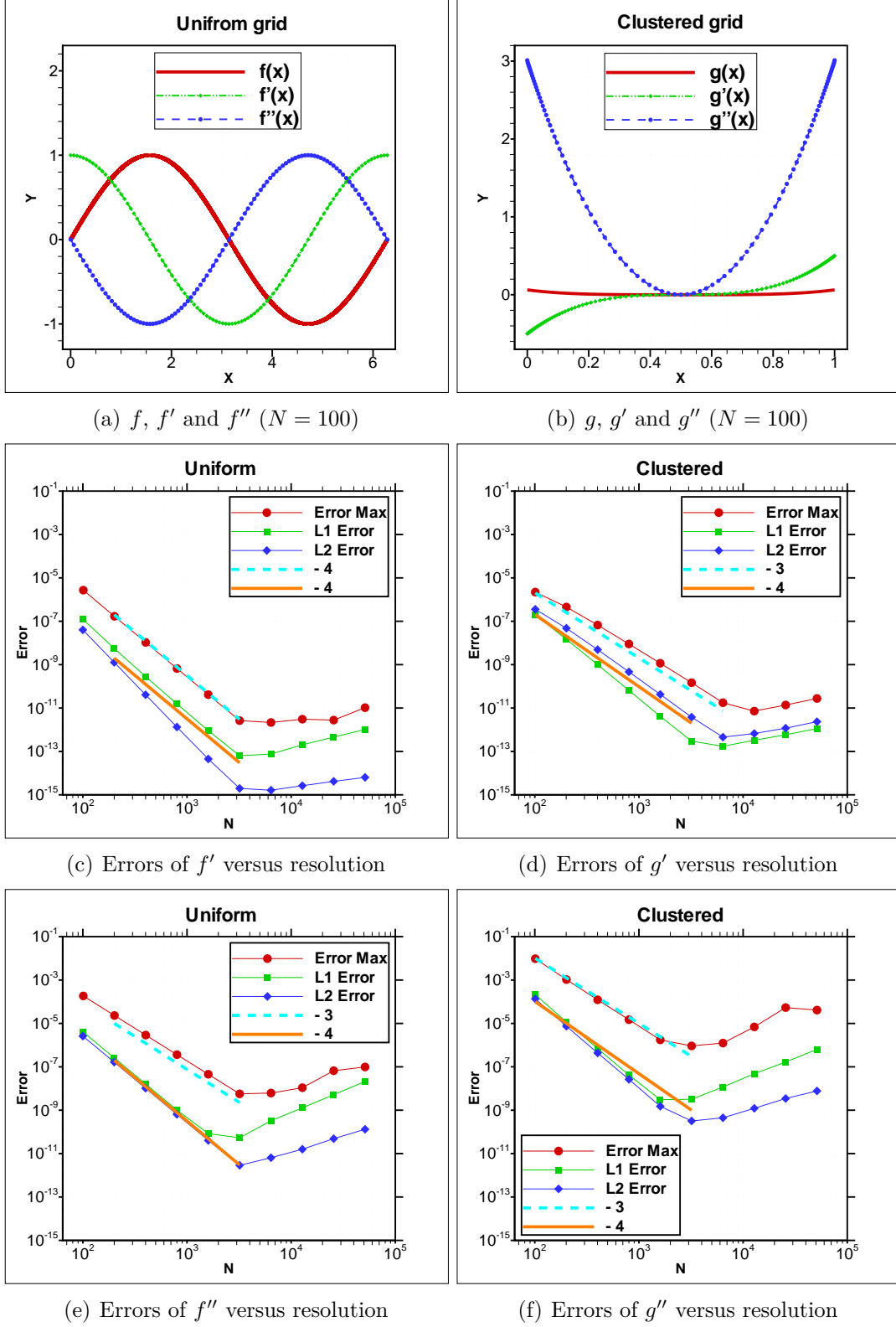


Figure 2.4: Truncation error analysis for first (c-d) and second (e-f) derivatives via a fourth-order compact method for $f(x) = \sin x, x \in [0, 2\pi]$, on a uniform grid (a), and $g(x) = (x - 0.5)^4, x \in [0, 1]$, on a clustered grid near boundaries (b) with Eq. (2.38), where $\gamma = 4$ and $\beta = \xi_{\max}/2$.

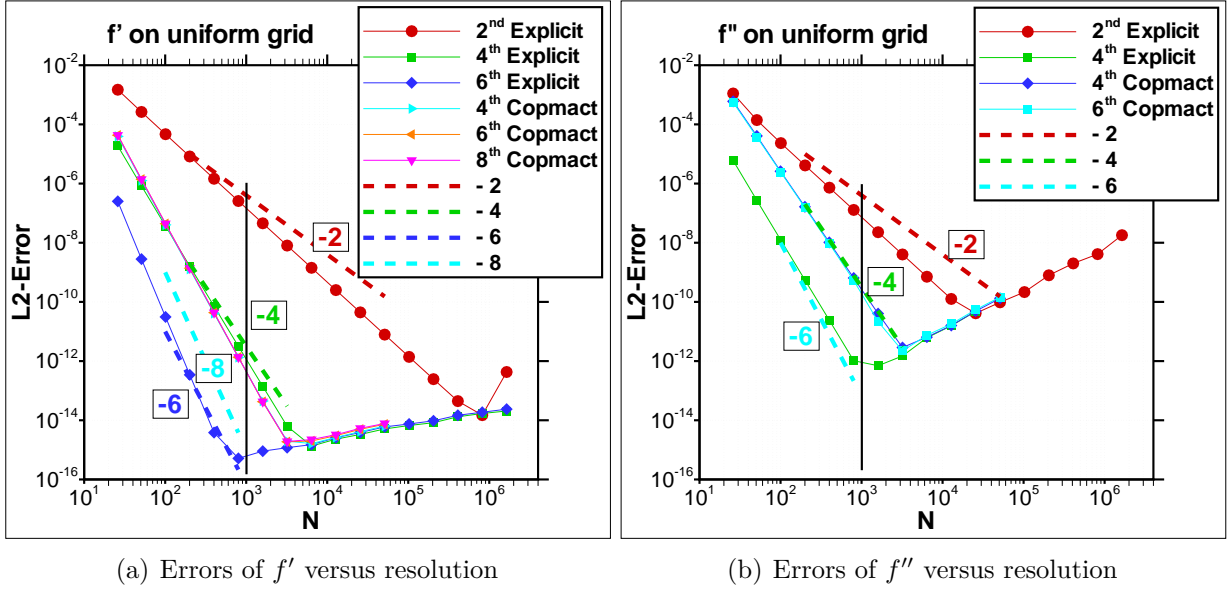


Figure 2.5: Truncation error analysis for first (a) and second (b) derivatives via different explicit and compact methods for $f(x) = \sin x$, $x \in [0, 2\pi]$, on a uniform grid.

Consider a single Fourier mode, namely $f(x) = e^{ikx}$ on a domain $x \in [0, 2\pi]$ and a uniform mesh of N points with spacing $\Delta x = 2\pi/N$ for discretization. The analytical first derivative of f at the j^{th} node is $f'(x_j)_{\text{exact}} = ik e^{ikx_j}$. The numerically computed derivative has a form like $f'(x_j)_{\text{FD}} = ik' e^{ikx_j}$, where k' is called the modified wavenumber for the first derivative which is a function of k and Δx . The difference between k' and k provides the differentiation error as a function of the resolution of the wave [66]. It is convenient to introduce a scaled wavenumber $w = k\Delta x$, where $w \in [0, \pi]$. Using the complex Fourier transform (cf. Appendix D) the exact first derivative of f in the Fourier space is $\hat{f}'_k|_{\text{exact}} = iw \hat{f}_k$. By applying different central finite difference schemes, it may be shown that $\hat{f}'_k|_{\text{FD}} = iw' \hat{f}_k$, where the scaled modified wavenumber $w' = k' \Delta x$ is real-valued and the nature of error is purely dispersive, i.e., central differencing schemes have no numerical diffusion (dissipation=0) for the first derivative. Spectral methods provide $w' = w$ for $w \in [0, \pi[$. For other methods, we have

$$w'(w) = \frac{-i \hat{f}'_k|_{\text{FD}}}{\hat{f}_k|_{\text{exact}}}, \quad k = 0, 1, 2, \dots, N/2$$

Taking the Fourier transform (cf. Appendix D) of Eq. (2.31) and through the use of Euler's formula ($e^{ix} = \cos x + i \sin x$) the following equation can be derived for the scaled modified wavenumber of the first derivative:

$$w'(w) = \frac{(a) \sin(w) + (b/2) \sin(2w) + (c/3) \sin(3w)}{1 + 2\alpha \cos(w) + 2\beta \cos(2w)} \quad (2.39)$$

For the second-order central explicit finite-difference one obtains $w' = k' \Delta x = \sin(k \Delta x)$. Fig. 2.6 shows the modified wavenumbers obtained by applying different central explicit

and implicit compact finite difference methods with $N = 2^{10}$ and $N = 2^6$ grid points. As can be seen, an insufficient number of grid points affects the ability of the method in dealing with high wavenumber oscillations. The corresponding resolution error for the first derivative

$$\epsilon(w) = \left| \frac{w'(w) - w}{w} \right|$$

is plotted in Fig. 2.7 for two pentadiagonal implicit compact methods. The error analysis for the second derivative is similar to that of the first derivative. The exact second derivative of the considered function generates a function with Fourier coefficients $\hat{f}_k''|_{\text{exact}} = -w^2 \hat{f}_k$. The numerical approximations via (2.32) correspond to $\hat{f}_k''|_{FD} = -w'' \hat{f}_k$, where

$$w''(w) = \frac{(2a)(1 - \cos(w)) + (b/2)(1 - \cos(2w)) + (2c/9)(1 - \cos(3w))}{1 + 2\alpha \cos(w) + 2\beta \cos(2w)} \quad (2.40)$$

is real-valued and the nature of the error is purely dissipative. The coefficients of different methods are given in Tables A.1 - A.5 (Appendix A). For the second-order central explicit finite-difference one obtains $w'' = 2(1 - \cos w)$. The difference between $w''(w)$ and w^2 is a measure of error in the second derivative approximation. Plots of $w''(w)$ versus scaled wavenumber for two explicit (second and fourth order) and two compact (tridiagonal fourth and sixth order) methods are presented in Fig. 2.8, with $N = 2^{10}$ and $N = 2^6$ grid points. A comparison of numerical estimation and analytical values given by Eq. (2.40) shows a good agreement between analytical and numerical estimations of scaled modified wavenumber. In addition to insufficient resolution, the forward/backward stencil used in high-order compact methods can also create some discrepancies between numerical and analytical analyses which are based on Fourier analysis and periodic boundary conditions. The corresponding resolution errors for the second derivative

$$\epsilon(w) = \left| \frac{w''(w) - w^2}{w^2} \right|$$

are plotted in Fig. 2.9.

2.2.4 Interpolation and filtering

Given the values of a function f on a uniformly spaced mesh $x_i = (i - 1)h$, ($i = 1, \dots, N$) where $h = L_x/(N - 1)$, interpolation can be performed with an approximation of the form [55]

$$\beta \hat{f}_{i-2} + \alpha \hat{f}_{i-1} + \hat{f}_i + \alpha \hat{f}_{i+1} + \beta \hat{f}_{i+2} = a \frac{f_{i+1/2} + f_{i-1/2}}{2} + b \frac{f_{i+3/2} + f_{i-3/2}}{2} + c \frac{f_{i+5/2} + f_{i-5/2}}{2} \quad (2.41)$$

where \hat{f}_i represents the interpolated value at node x_i , explicit and implicit schemes of different formal accuracy may be derived by matching the Taylor series coefficients of

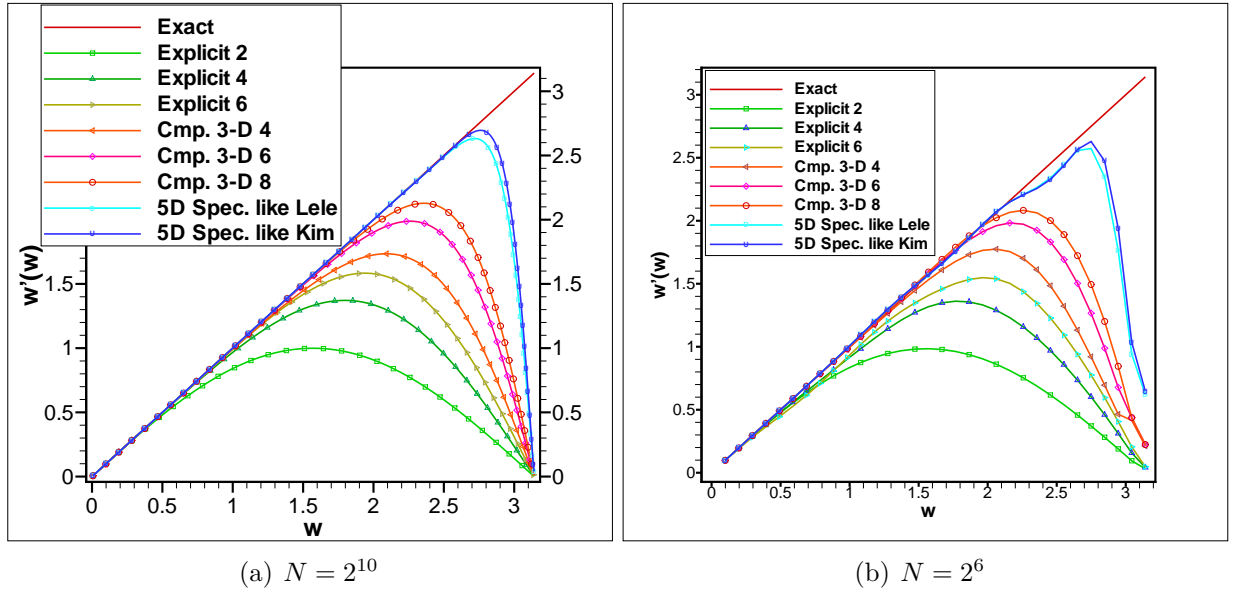


Figure 2.6: Plots of the scaled modified wavenumber $w'(w)$ versus the scaled wavenumber $w = k\Delta x$ for the first derivative with the use of different central finite difference methods for $f(x) = e^{ikx}$ and $x \in [0, 2\pi]$.

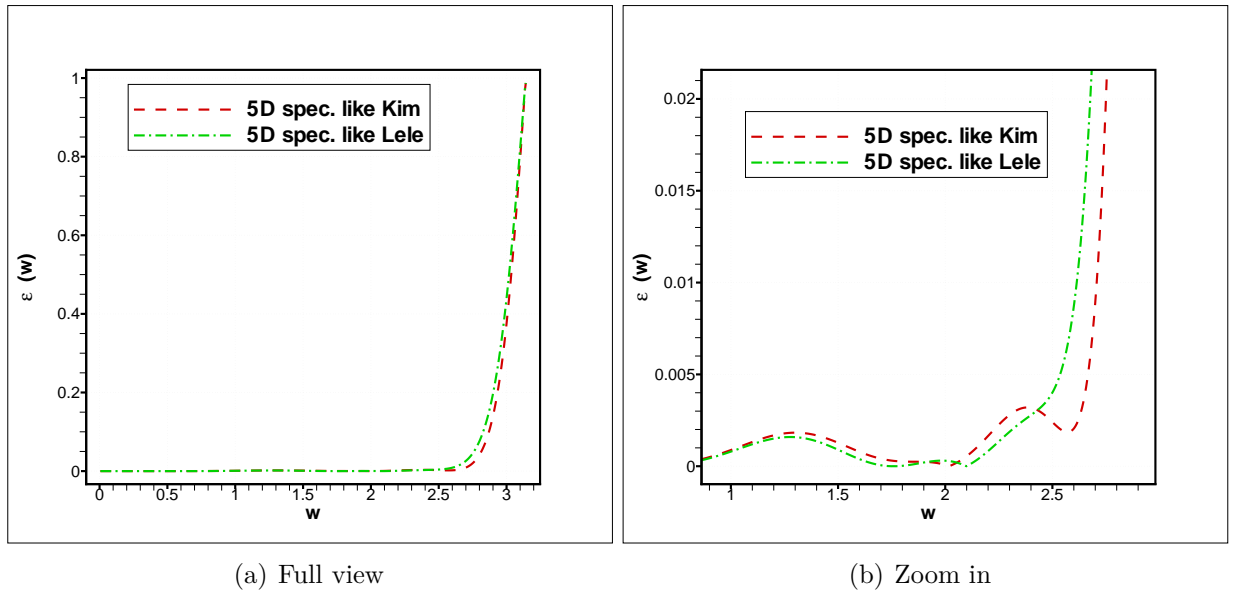


Figure 2.7: Plots of the resolution error $\epsilon(w) = \left| \frac{w'(w) - w}{w} \right|$ for the first derivative via two pentadiagonal spectral like compact method proposed by Lele [55] and Kim [111], with $N = 2^{10}$ grid points.

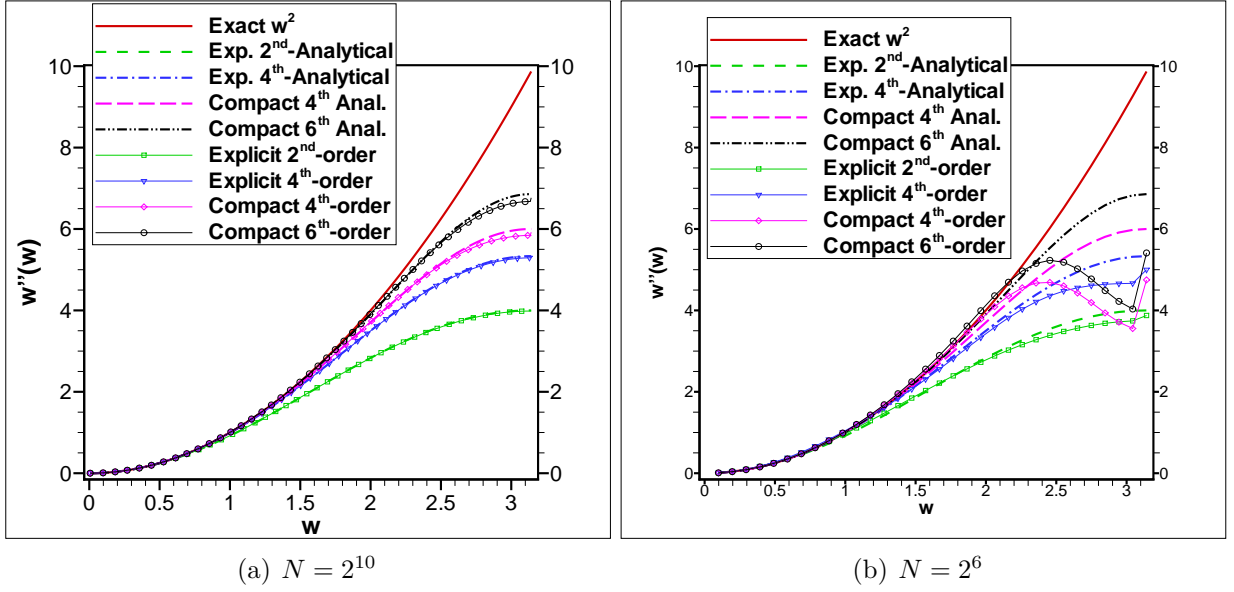


Figure 2.8: Plots of the scaled modified wavenumber $w''(w)$ versus the scaled wavenumber $w = k\Delta x$ for the second derivative with the use of different central finite difference methods for $f(x) = e^{ikx}$ and $x \in [0, 2\pi]$.

various order. The transfer function associated with Eq. (2.41) is

$$T(w) = \frac{a \cos(w/2) + b \cos(3w/2) + c \cos(5w/2)}{1 + 2\alpha \cos(w) + 2\beta \cos(2w)}. \quad (2.42)$$

Explicit methods are constructed with $\alpha = \beta = 0$. First order interpolation can be constructed with $(b = c = 0)$ and $a = 1$. A third-order explicit interpolation is obtained with $c = 0$, $a = 9/8$ and $b = -1/8$. At the left and right boundaries forward and backward interpolations must be used. With a $1/2$ shift in the indices a third-order approximation can be found with

$$\hat{f}_{1/2} = (5f_0 + 15f_1 - 5f_2 + f_3)/16 \quad (2.43)$$

$$\hat{f}_{N-1/2} = (5f_N + 15f_{N-1} - 5f_{N-2} + f_{N-3})/16 \quad (2.44)$$

and a second-order approximation near boundaries can be made with

$$\hat{f}_{1/2} = (6f_0 + 12f_1 - 2f_2)/16 \quad (2.45)$$

$$\hat{f}_{N-1/2} = (6f_N + 12f_{N-1} - 2f_{N-2})/16 \quad (2.46)$$

The implicit fourth-order interpolations are defined by

$$a = (9 + 10\alpha - 14\beta + 16c)/8 \quad \text{and} \quad b = (-1 + 6\alpha + 30\beta - 24c)/8$$

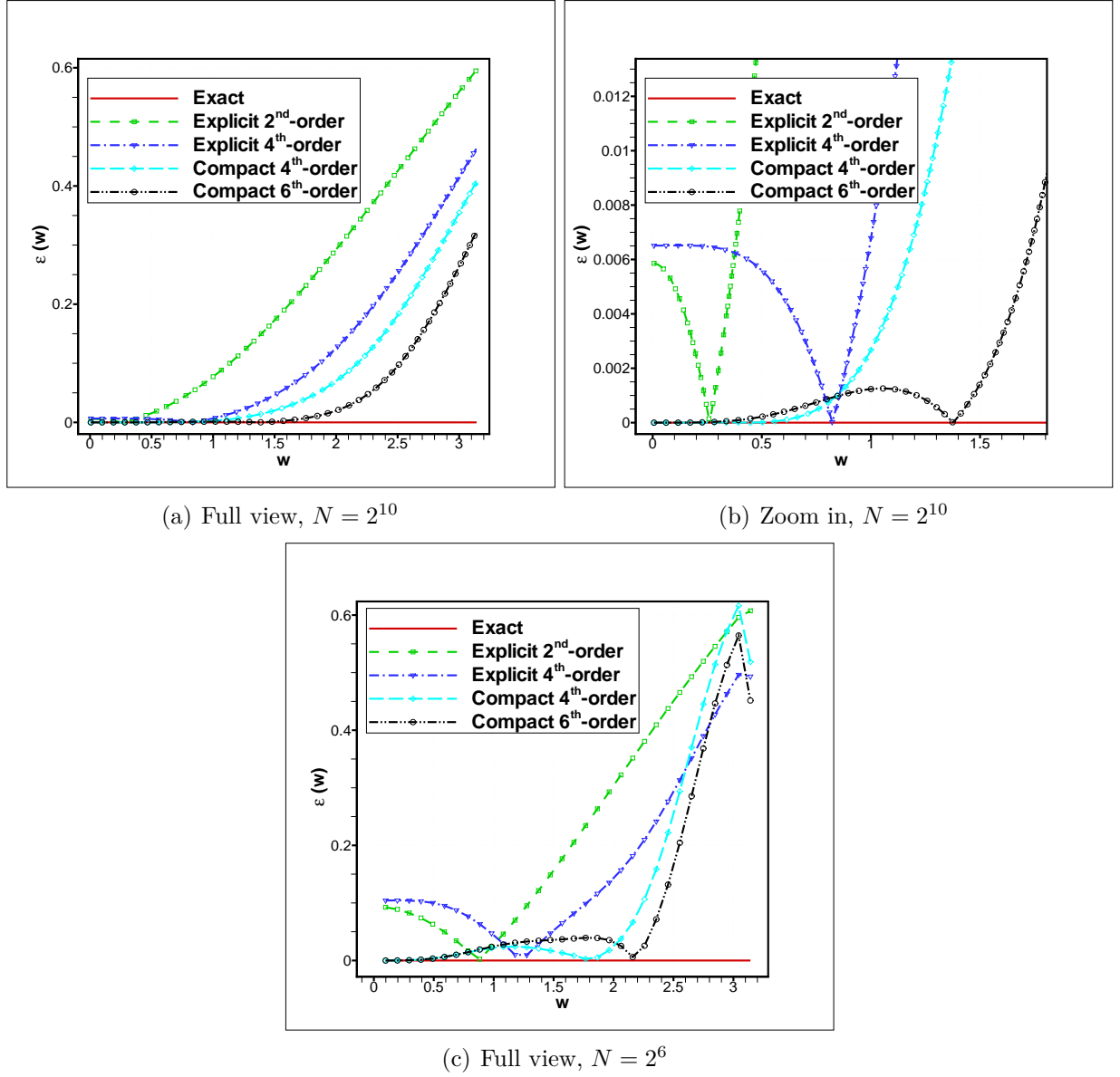


Figure 2.9: Plots of the resolution error $\epsilon(w) = \left| \frac{w''(w) - w^2}{w^2} \right|$, for the second derivative via different methods.

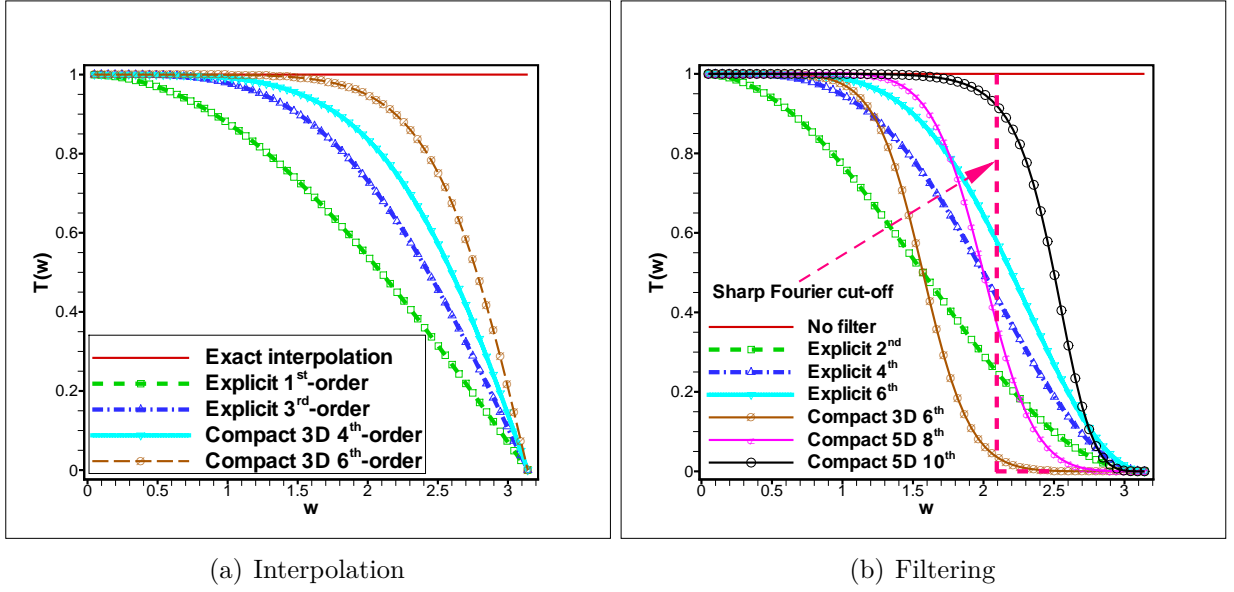


Figure 2.10: Transfer functions $T(w)$ versus scaled wavenumber for different explicit and compact tridiagonal/pentadiagonal methods for interpolation and filtering.

An implicit fourth-order approximation can be defined by $\alpha = 1/6$, $\beta = 0$, $a = 4/3$ and $b = c = 0$. A six-order approximation can be defined by $\alpha = 3/10$, $\beta = 0$, $a = 3/2$, $b = 1/10$ and $c = 0$. Near boundaries, a forward/backward fourth-order approximation can be used with a $1/2$ shift in the index, e.g.,

$$\hat{f}_{1/2} + \hat{f}_{3/2} = \frac{1}{2} \frac{f_0 + f_2}{2} + \frac{3}{2} f_1 + O(h^4) \quad (2.47)$$

we refer to Lele [55] for more details. The transfer function $T(w)$ versus modified wavenumber, given by Eq. (2.42) for different explicit and compact tridiagonal methods is plotted in Fig. 2.10 (a). As can be seen the interpolation eliminates high wavenumber oscillations (corresponding to small scales) of the original function.

High wavenumber oscillations can be seen in the flow field due to different reasons, e.g., insufficient resolution, central differencing and collocated grid for the pressure and the velocity fields. For elimination of high wavenumber oscillations in the flow variables a common practice is low-pass filtering. Compact filtering is motivated in consistency with compact differentiation and is done with an approximation of the form

$$\beta \bar{f}_{i-2} + \alpha \bar{f}_{i-1} + \bar{f}_i + \alpha \bar{f}_{i+1} + \beta \bar{f}_{i+2} = a f_i + b \frac{f_{i+1} + f_{i-1}}{2} + c \frac{f_{i+2} + f_{i-2}}{2} + d \frac{f_{i+3} + f_{i-3}}{2} \quad (2.48)$$

where \hat{f}_i represents the filtered values at the node x_i . With adjusting the coefficients different filters can be constructed. The application of the filtering discussed here is removing

short length scales in physical space. The transfer function associated with Eq. (2.48) is

$$T(w) = \frac{a + b \cos(w) + c \cos(2w) + d \cos(3w)}{1 + 2\alpha \cos(w) + 2\beta \cos(2w)}. \quad (2.49)$$

Explicit filters are constructed with $\alpha = \beta = 0$, e.g., by choosing $a = \beta_s/(2 + \beta_s)$ and $b = 2/(2 + \beta_s)$ we obtain a simple filter which is known as Shuman filter [15]

$$\bar{f}_i = \frac{f_{i-1} + \beta_s f_i + f_{i+1}}{2 + \beta_s} \quad (2.50)$$

in which $2 \leq \beta_s \leq 100$. This filter is typically used to mollify the mask (characteristic) function when using the penalization method, if a high-order scheme is used to compute derivatives of the spatial terms in the Navier–Stokes equations. By choosing $\beta_s = 2$, hence $a = b = 0.5$, we obtain

$$\bar{\chi}_i = \frac{\chi_{i-1} + 2\chi_i + \chi_{i+1}}{4} \quad (2.51)$$

the two-dimensional version of the Shuman filter is

$$\bar{\chi}_{ij} = \frac{2\chi_{ij} + \sum_{nb=1}^4 \chi_{nb}}{6} \quad (2.52)$$

for which $T(w) = 0.5(1 + \cos w)$, usually the values of the filtered variable at the boundaries remain unchanged in the process of filtering in physical space. Following [79] and [122] in the present study the mask function introduced in the context of penalization is mollified by the two-dimensional Shuman [15] filter, which is equivalent to raised *cosine* filter in Fourier space. We refer to [121] for more details. The effect of smoothing with the one-dimensional Shuman filter Eq. (2.51) is demonstrated in Fig. 2.11 for a box function $\chi(x) = 1, x \in [4.5, 5.5]$ which is represented by the red-solid line. The mollified function $\bar{\chi}$ is represented with the green-dashed line.

When discretizing the Navier–Stokes equations with high-order central collocated finite-differences, due to insufficient resolution, high wavenumber oscillations can occur in the flow field. In this case implicit high-order filters (at least two order higher than formal accuracy of the solution itself) are used in order to filter the solution at each time-step for stabilizing the numerical simulation. Tridiagonal implicit filters of the form

$$\alpha \bar{f}_{i-1} + \bar{f}_i + \alpha \bar{f}_{i+1} = \sum_{n=0}^N \frac{a_n}{2} (f_{i+n} + f_{i-n}) \quad (2.53)$$

will be used sequentially in each spatial direction, where α is a free parameter in the range $0 < |\alpha| \leq 0.5$. Special treatment is necessary at near boundary points due to the relatively large stencil of the filter. Two approaches can be applied near the boundary points, either, the order of accuracy, thus the stencil, can be reduced when coming close to the boundary

to a level for which a central scheme is available, or one-side forward-backward stencil with slightly reducing the order of accuracy can be used. When the mesh is highly clustered near the boundary, the former approach is more stable and will thus be preferred. We refer to the original paper by Lele [55] and also the paper by Visbal and Gaitonde [85] for the coefficients and more discussions. The transfer functions versus scaled wavenumber $T(w)$, given by Eq. (2.49) for different explicit and compact tridiagonal/pentadiagonal methods are plotted in Fig. 2.10 (b). As can be seen the filtering eliminates high wavenumber oscillations (corresponding to small scales) of the original function. Moreover, it is clear that compact filters are sharper in Fourier space than explicit filters and with using high-order filters the efficiency for eliminating higher wavenumbers is increased. A comparison of compact filters with a Fourier cut-off filter shows that it is difficult to construct a sharp filter in physical space.

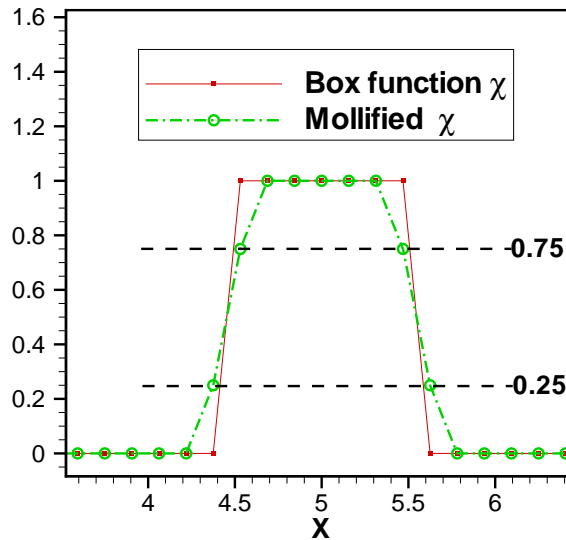


Figure 2.11: A box function $\chi(x) = 1$, $x \in [4.5, 5.5]$ (red-solid) and a mollified box function $\bar{\chi}$ (green-dashed) with Eq. (2.51).

2.3 Spatial discretization of the Poisson equation

In the procedure of numerical solution of the incompressible flows a Poisson equation, e.g., Eq. (2.6), must be solved to ensure the incompressibility constraint. This is the most time consuming part of the algorithm due to the elliptic characteristic of the Poisson equation. Thus an efficient approach can considerably enhance the computational efficiency. A wide variety of methods is available in the literature which can be divided into direct and iterative methods. We will present an iterative and a direct method for second and fourth order discretizations of the Poisson equation. According to the boundary conditions different

methods can be used. However FFT-based direct methods put some constraints at the boundaries.

2.3.1 An iterative method for solution of the Poisson equation

The two dimensional Poisson equation (2.6) in Cartesian coordinates reads

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \quad (2.54)$$

Using a uniform ($\Delta x = \Delta y = h$) Cartesian grid like the one illustrated in Fig. 2.1, for discretization of the solution domain and replacing the derivatives with second-order central finite differences we have

$$\delta_x^2 \psi + \delta_y^2 \psi = -\omega_{ij} \quad (2.55)$$

where δ_x^2 and δ_y^2 are central second-order finite-difference approximations of the second derivatives, e.g.,

$$\delta_x^2 \psi_{i,j} = \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{h^2} + O(h^2) \quad (2.56)$$

In the y -direction a same formula can be derived. With the use of Dirichlet boundary conditions on the discrete solution domain, these algebraic equations form a linear system of equations

$$\begin{bmatrix} \ddots & & & & & \\ \ddots & \ddots & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ A_S & & & A_W & A_P & A_E & & A_N \\ & \ddots & & & & & \ddots & \\ & & \ddots & & & & & \ddots \\ & & & \ddots & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \psi_{i,j-1} \\ \vdots \\ \vdots \\ \vdots \\ \psi_{i-1,j} \\ \psi_{i,j} \\ \psi_{i+1,j} \\ \vdots \\ \vdots \\ \vdots \\ \psi_{i,j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ -h^2 \omega_{i,j} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

where

$$A_P = -4 \quad (2.57)$$

$$A_E = A_W = A_N = A_S = 1 \quad (2.58)$$

The linear system of equations can be solved via an iterative method, namely the point successive over relaxation (PSOR) method

$$\psi_{i,j}^{n+1} = \beta \psi_{i,j}^{\text{new}} + (1 - \beta) \psi_{i,j}^{\text{old}} \quad (2.59)$$

where β is the over relaxation factor, $\beta \in [1, 2]$. Thereby we have

$$\psi_{i,j}^{n+1} = \psi_{i,j}^{\text{old}} + \beta \text{Res}(i, j) \quad (2.60)$$

By the use of the Gauss-Seidel method we have

$$\text{Res}(i, j) = 0.25 \left(\sum_{nb=1}^4 \psi_{nb}^{\text{last}} - 4\psi_{i,j}^{\text{old}} + h^2 \omega_{i,j} \right) \quad (2.61)$$

where $h = \Delta x = \Delta y$ and nb denotes the neighbor points (by applying a red-black sweeper always the updated values of the neighbors will be used). The norm of the residual $\|Res\|$, must converge below a prescribed convergence criterion, i.e.,

$$\|Res\| \leq \max(\epsilon \Delta x^n, \epsilon_{\text{machine}})$$

where n is the order of discretization (for instance, $n = 2$ or 4), ϵ is case dependent, e.g., $\epsilon \in [10^{-6}, 10^{-4}]$ and $\epsilon_{\text{machine}}$ is the rounding² error. See Table 2.1 for the values of residuals and Table 2.2 for the number of iterations necessary to satisfy the convergence criterion, with a second and fourth order elliptic solver, for the considered problem presented in Fig. 2.12.

Fourth-order approximation of the Poisson equation (2.54) can be obtained by evaluation of each term with the following expressions:

$$\frac{\partial^2 \psi}{\partial x^2} = \delta_x^2 \psi - \frac{\Delta x^2}{12} \frac{\partial^4 \psi}{\partial x^4} + O(\Delta x^4) \quad (2.62)$$

$$\frac{\partial^2 \psi}{\partial y^2} = \delta_y^2 \psi - \frac{\Delta y^2}{12} \frac{\partial^4 \psi}{\partial y^4} + O(\Delta y^4) \quad (2.63)$$

To obtain a fourth-order finite-difference compact formulation, the correction term, i.e., the $O(\Delta x^2, \Delta y^2)$ terms in Eqs. (2.62) and (2.63) cannot be dropped and must be evaluated. Because of the presence of Δx^2 and Δy^2 factor behind high-order derivatives, these terms can be approximated with second-order accuracy. Hence the whole approximation scheme yields the fourth-order accuracy. These approximation formulas are well-known and have been analyzed by Collatz in [17] and are equivalent to the compact schemes discussed by Lele [55]. Following Spitz et al. [61] high-order derivatives can be found by using the

²According to IEEE 754-2008 standard $\epsilon_{\text{machine}}$ is of order 10^{-7} for single precision and 10^{-16} for double precision computations.

original Poisson equation (2.54) and simply successive differentiating with respect to x and y directions,

$$\frac{\partial^4 \psi}{\partial x^4} = -(\psi_{yyxx} + \omega_{xx}) \quad (2.64)$$

$$\frac{\partial^4 \psi}{\partial y^4} = -(\psi_{xxyy} + \omega_{yy}) \quad (2.65)$$

where subscripts indicate partial differentiation (i.e, $\psi_{xx} = \partial_{xx}\psi$). By replacing the above equations in (2.62) and (2.63) we have:

$$\frac{\partial^2 \psi}{\partial x^2} = \delta_x^2 \psi + \frac{\Delta x^2}{12}(\psi_{yyxx} + \omega_{xx}) + O(\Delta x^4) \quad (2.66)$$

$$\frac{\partial^2 \psi}{\partial y^2} = \delta_y^2 \psi + \frac{\Delta y^2}{12}(\psi_{xxyy} + \omega_{yy}) + O(\Delta y^4) \quad (2.67)$$

Substituting in the original Poisson Eq. (2.54) yields,

$$\delta_x^2 \psi + \frac{\Delta x^2}{12}(\psi_{yyxx} + \omega_{xx}) + \delta_y^2 \psi + \frac{\Delta y^2}{12}(\psi_{xxyy} + \omega_{yy}) = -\omega \quad (2.68)$$

By considering $\Delta x = \Delta y = h$ we obtain:

$$(\delta_x^2 + \delta_y^2 + \frac{h^2}{6}\delta_x^2\delta_y^2)\psi = -\omega - \frac{h^2}{12}(\delta_x^2 + \delta_y^2)\omega \quad (2.69)$$

By imposing Dirichlet boundary conditions on the discrete solution domain, these algebraic equations form a linear system of equation of the form,

$$A\psi = B \quad (2.70)$$

where the matrix of the coefficients A (on the left-hand side) is a nine-diagonal band matrix. Depending on the row-wise or column-wise arrangement of the unknowns the matrix will be slightly different. If one chooses the row-wise arrangement, a coefficient matrix of the form illustrated in Appendix C will be obtained. This book-keeping matrix can be illustrated in a symbolic manner via Eq. (2.77), which represents the coefficients corresponding to stencil of a given point (i, j) . Considering Fig. 2.12 (c) the coefficients of each point are:

$$A_P = -20 \quad (2.71)$$

$$A_E = A_W = A_N = A_S = 4 \quad (2.72)$$

$$A_{NE} = A_{NW} = A_{SE} = A_{SW} = 1 \quad (2.73)$$

Table 2.1: The considered convergence criteria for the residual $\|Res\|$ of the Poisson solvers, for the problem $(\nabla^2 u = f)$ presented in Fig. 2.12, using different resolutions.

Grid	PSOR 2nd-order	PSOR 4th-order
33^2	4×10^{-7}	1.5×10^{-9}
65^2	1×10^{-7}	9×10^{-11}
129^2	2×10^{-8}	6×10^{-12}
257^2	6×10^{-9}	4×10^{-13}
513^2	1.5×10^{-9}	2×10^{-14}
1025^2	4×10^{-10}	1×10^{-15}
2049^2	1×10^{-10}	1×10^{-15}
4097^2	2×10^{-11}	1×10^{-15}

where P is used to represent the point (i, j) , for E, W, N, S and other neighbors, we refer again to Fig. 2.12 (c). The right hand side is given by

$$B = -\frac{h^2}{2}(12 + h^2\delta_x^2 + h^2\delta_y^2) \omega \quad (2.74)$$

thus in discrete form we have

$$B_{i,j} = -\frac{h^2}{2}(8\omega_{i,j} + \omega_{i+1,j} + \omega_{i-1,j} + \omega_{i,j+1} + \omega_{i,j-1}) \quad (2.75)$$

The final discretized equation can be written symbolically as:

$$\begin{pmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{pmatrix} \psi = -\frac{h^2}{2} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 8 & 1 \\ 0 & 1 & 0 \end{pmatrix} \omega \quad (2.76)$$

or

$$\begin{pmatrix} A_{SW} & A_S & A_{SE} \\ A_W & A_P & A_E \\ A_{NW} & A_N & A_{NE} \end{pmatrix} \psi = B \quad (2.77)$$

The linear system of equations (2.77) can be solved using the Gauss-Seidel method Eq. (2.60) with

$$Res(i, j) = 0.05 \left(\sum_{nb=1}^8 \psi_{nb}^{\text{last}} - 20\psi_{i,j}^{\text{old}} - B_{i,j} \right) \quad (2.78)$$

See Fig. 2.12 (d) for an error analysis of the Poisson solvers via the PSOR method for second-order and fourth-order compact discretization.

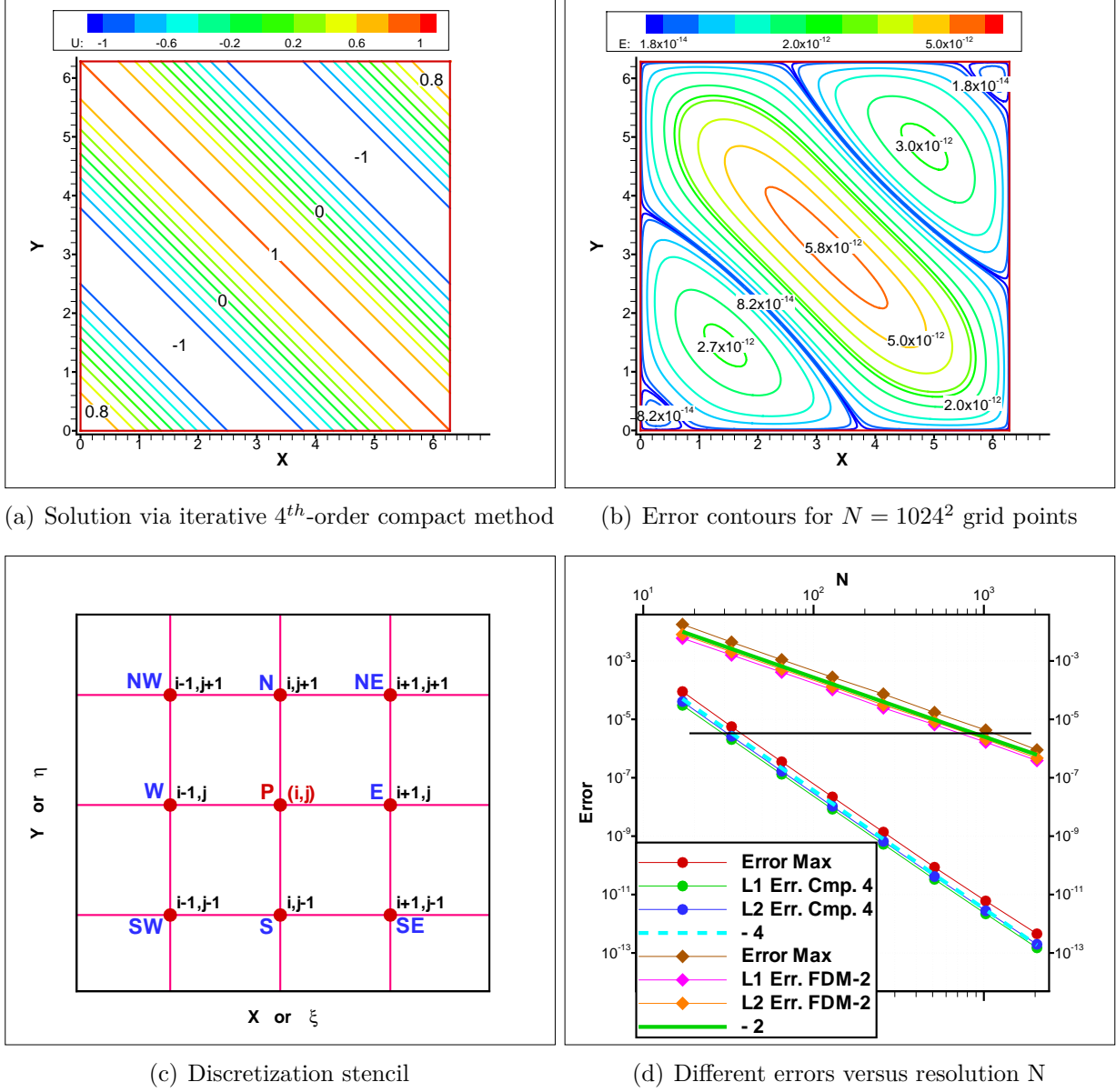


Figure 2.12: (a) Solution of the Poisson equation ($\nabla^2 u = f$) with the forcing term $f(x, y) = -2 \cos(x + y)$, $(x, y) \in \Omega = [0, 2\pi]^2$ and Dirichlet boundary conditions $u_b(x, y) = \cos(x + y)$, (x, y) on $\partial\Omega$ via an iterative fourth-order compact method. (b) Corresponding error contours $E(x, y) = |u(x, y) - \cos(x + y)|$, $(x, y) \in \Omega$ in comparison with the exact solution for $N = 1024^2$ grid points. (c) The 9-point stencil used in the fourth-order compact discretization of the Poisson equation on a two-dimensional grid. (d) Error analysis for Poisson solvers via the PSOR method using second and fourth order compact discretizations.

Table 2.2: The number of iterations of the Poisson ($\nabla^2 u = f$) solvers, for the problem presented in Fig. 2.12, using different resolutions.

Grid	PSOR 2nd-order	PSOR 4th-order
33^2	618	926
65^2	673	1064
129^2	718	1204
257^2	772	1342
513^2	2805	4571
1025^2	12017	21057
2049^2	48790	82042
4097^2	more than 100000	more than 300000

2.3.2 A high-order fast Poisson solver

In solving the incompressible Navier–Stokes equations an elliptic Poisson equation is frequently encountered which is the most time consuming part of the algorithm. The common case is the pressure Poisson equation normally used with homogeneous Neumann boundary conditions, for the pressure correction in projection methods. In the vorticity-stream function formulation, Eq. (2.6) has to be solved with Dirichlet boundary conditions for vorticity and stream-function. Free slip boundary conditions in a closed rectangular domain ($\omega = \psi = 0$; at all boundaries) is applied in all test cases studied in the present investigation. Numerical tests reveal that there is no significant difference between no-slip and free-slip boundary conditions, for fluid structure interaction problems, if the size of solution domain is big enough in comparison to the body length, we refer to the discussion in Section 3.4.1. In the presence of periodic boundary conditions, FFT based direct solvers can be used to efficiently solve the Poisson equation with high accuracy. Even if the flow is not periodic in all directions, like most of practical problems, in accordance with the boundary conditions for the elliptic equation (homogeneous Dirichlet/Neumann) *sine* or *cosine* FFTs can be used in one or two directions, see Fig. 2.13 and [44], [77] and [131].

Here we are presenting a new direct fourth-order solver for the Poisson equation (2.6) which is a combination of compact finite differences with a sine FFT for alternative directions. This method is suitable for imposing free-slip boundary condition at least in one direction. The advantages of our method are fourth-order accuracy, convergence down to machine zero by using an optimal grid, compact tridiagonal stencils, possibility of extension to three dimensions, reduced arithmetics and memory usage in comparison to iterative methods. Moreover, the parallelization is straightforward because of decoupling the operations in different directions. Nearly linear strong scaling (speed up) and efficiency is reported by Laizet and Lamballais [131] for a direct solver by decoupling of the operators in alternative directions. They introduced a dual domain decomposition (or pencil) method, in

which information along a line is accessible for a CPU by alternative decomposition of the domain in three directions, see Fig. 6.2. The limitation of our method (in addition to the boundary conditions) is the use of a uniform grid in the direction in which the FFT is applied. When the solver of the parabolic part is based on finite-differences, it is a custom to use a FDM discretization in one direction without loss of accuracy and efficiency via a direct tridiagonal solver. The advantage of using finite differences in one direction is the possibility of applying general boundary conditions and using a refined mesh.

For a second-order collocated discretization of the 3D Poisson equation ($\nabla^2 u = f$) by applying Fourier transforms in y and z directions and by replacing second derivatives with $-k^2 \hat{u}_{i,m,n}$ in Fourier space, we get

$$\delta_x^2 \hat{u}_{i,m,n} - (k_y^2 + k_z^2) \hat{u}_{i,m,n} = \hat{f}(i, m, n) \quad (2.79)$$

Usually the exact wavenumber is replaced by modified wavenumber k' which permits to evaluate the difference between the finite-difference and the spectral approximation of a second derivative [77]. For a second-order explicit finite-difference discretization with the use of Table A.1 by choosing ($a = 1$ and $\alpha = \beta = b = c = 0$) and replacing in Eq. (2.40) the modified wavenumbers of each direction can be obtained as follows:

$$k_y'^2 = \frac{2}{\Delta y^2} \left(1 - \cos(k_y \pi / J_{max}) \right) \quad , \quad k_z'^2 = \frac{2}{\Delta z^2} \left(1 - \cos(k_z \pi / K_{max}) \right)$$

see Fig. 2.8 for modified wavenumbers of the second derivative obtained for a second-order finite-difference discretization. The final tridiagonal system to be solved (see Appendix F) for the solution in Fourier space for each wavenumber is [116]

$$\boxed{\hat{u}_{i+1,m,n} - (2 + h^2 k_y'^2 + h^2 k_z'^2) \hat{u}_{i,m,n} + \hat{u}_{i-1,m,n} = h^2 \hat{f}(i, m, n)} \quad (2.80)$$

where $h = \Delta x$. More details can be found in [77].

For a compact fourth-order collocated discretization of the 2D Poisson equation $-\nabla^2 \psi = \omega$, over $N_x \times N_y$ grid points, by using

$$\frac{\partial^2 \psi}{\partial x^2} = \delta_x^2 \psi - \frac{\Delta x^2}{12} \frac{\partial^4 \psi}{\partial x^4} + O(\Delta x^4) \quad (2.81)$$

where δ_x^2 represents a central second-order approximation of the second derivative, for the x -direction we obtain

$$\left(\delta_x^2 - \frac{\Delta x^2}{12} \frac{\partial^4}{\partial x^4} + \partial_{yy} \right) \psi = -\omega \quad (2.82)$$

because of the presence of the Δx^2 factor behind the fourth-order derivative, this term cannot be dropped and must be evaluated with second-order accuracy. Therefore, the whole approximation scheme yields fourth-order accuracy. The fourth-order derivative can be evaluated by using the original Poisson equation $-\nabla^2\psi = \omega$, and successive differentiation with respect to x (i.e., $\partial_{xx}\partial_{xx}\psi = -\partial_{xx}\partial_{yy}\psi - \partial_{xx}\omega$). Replacing ∂_{xx} by δ_x^2 , we find

$$(\delta_x^2 + \frac{\Delta x^2}{12}\delta_x^2\partial_{yy} + \partial_{yy})\psi = -\omega - \frac{\Delta x^2}{12}\delta_x^2\omega \quad (2.83)$$

By applying a Fourier transform in y direction in Eq. (2.83) and replacing second derivatives by $-k_y^2\hat{\psi}$ in Fourier space, we have

$$(\delta_x^2 - \frac{\Delta x^2}{12}\delta_x^2k_y'^2 - k_y'^2)\hat{\psi} = -\hat{\omega} - \frac{\Delta x^2}{12}\delta_x^2\hat{\omega} \quad (2.84)$$

Usually the exact wavenumber is replaced by the modified wavenumber $k_y'^2$ which permits to adapt the spectral approximation of the second derivative with the considered finite difference method [77]. For a fourth-order explicit finite difference discretization, in Table A.1 by choosing ($a = 4/3$, $b = -1/3$ and $\alpha = \beta = c = 0$) and replacing the coefficients in the analytical relation (2.40) given by Lele [55], the scaled modified wavenumber of the second derivative is given as follows:

$$k_y'^2 = \frac{1}{\Delta y^2} \left[\frac{8}{3} \left(1 - \cos\left(\frac{k_y\pi}{N_y}\right) \right) - \frac{1}{6} \left(1 - \cos\left(\frac{2k_y\pi}{N_y}\right) \right) \right] \quad (2.85)$$

Comparison with numerical values in Fig. 2.8 (b) confirms that Eq. (2.85) is indeed exact. The final tridiagonal system to be solved (see Appendix F) in Fourier space for each wavenumber of ψ in the y -direction is

$$\boxed{\beta\hat{\psi}_{i+1,m} - (2\beta + k_y'^2)\hat{\psi}_{i,m} + \beta\hat{\psi}_{i-1,m} = -(\hat{\omega}_{i+1,m} + 10\hat{\omega}_{i,m} + \hat{\omega}_{i-1,m})/12} \quad (2.86)$$

for $i = 2, \dots, N_x - 1$, where $\beta = \Delta x^{-2} - k_y'^2/12$. In summary, first a one-dimensional direct-FFT of the forcing term ω is performed along all lines in the y -direction. Then for each line in the x -direction the tri-diagonal system (2.86) must be solved to find the solution ψ in wavenumber space. Next an inverse-FFT of the solution is performed line by line in the y -direction. For real data with zero value at the boundaries (homogeneous Dirichlet, i.e., $\psi = \omega = 0$, corresponding to free-slip boundary conditions), the natural Fourier transform to use is the *sine* transform, see Fig. 2.13 from [56]. The direction of FDM and FFT can be changed to consider no-slip boundary condition in the y -direction. In order to take into account inflow/outflow boundary conditions the mean flow must be reduced from the total velocity field $\mathbf{u} = \mathbf{U} - \mathbf{U}_\infty$ in the vorticity transport equation (2.5) to impose $\psi = 0$ at the boundaries. This is equivalent to move the grid with \mathbf{U}_∞ and

writing the Navier–Stokes equations in a moving reference frame for the perturbed velocity field \mathbf{u} , instead of a Galilean inertial frame [135].

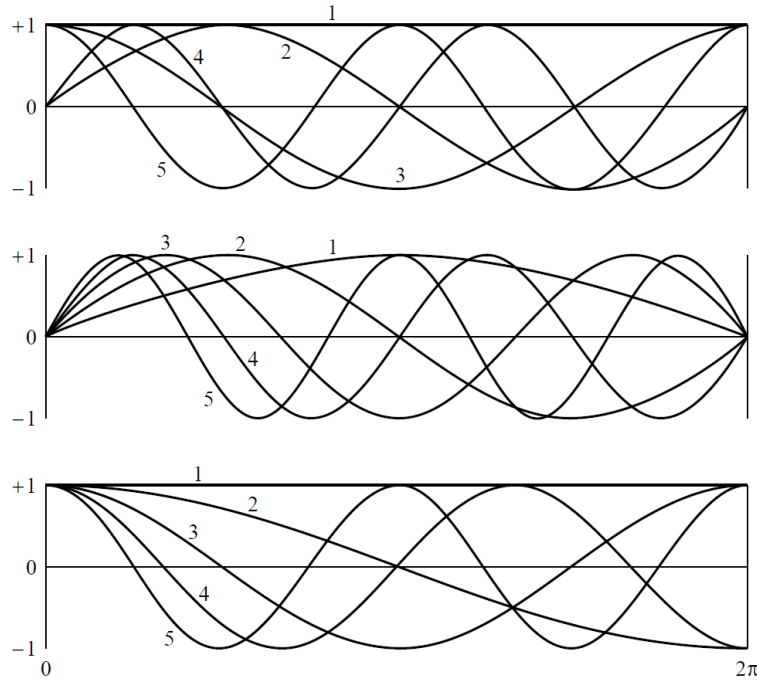


Figure 2.13: (top) The trigonometric basis functions for a complex FFT of a periodic function. (center) The trigonometric basis functions for a *sine* FFT of a function with homogeneous Dirichlet boundary conditions. (bottom) The trigonometric basis functions for a *cosine* FFT of a function with homogeneous Neumann boundary conditions. Picture from [56].

Validation of the fast Poisson solvers

For validation of the developed second and fourth order Poisson ($\nabla^2 u = f$) solvers the following analytical solution is considered

$$u_{\text{exact}}(x, y) = \cos(x) \sin(n\pi y/L_y) \quad , \quad (x, y) \in \Omega = [0, 2\pi]^2$$

with the corresponding forcing term

$$f(x, y) = -(n^2\pi^2/L_y^2 + 1) \cos(x) \sin(n\pi y/L_y) \quad , \quad (x, y) \in \Omega$$

and Dirichlet boundary conditions $u_b(x, y) = \cos(x) \sin(n\pi y/L_y)$, (x, y) on $\partial\Omega$. The solution for $n = 3$ with $N = 33^2$ grid points, computed with the fourth order direct solver is illustrated in Fig. 2.14 (a). The contours of solution error in comparison with the exact solution $E(x, y) = |u(x, y) - u_{\text{exact}}(x, y)|$, $(x, y) \in \Omega$ are illustrated in Fig. 2.14 (b). Different errors versus spatial resolution computed with second and fourth order compact direct solvers are illustrated in Fig. 2.14 (c). The CPU-times for different resolutions/methods are

Table 2.3: CPU-times (second / 100 CALL) for different (multigrid, point successive over relaxation and direct) Poisson solvers, for the problem ($\nabla^2 u = f$) presented in Fig. 2.14, using different resolutions. The processor is Intel(R) Core(TM) i5-3230M CPU@2.6 GHz.

Grid	PSOR 2	PSOR 4	MG 2	Direct 2	Direct 4
33^2	1.5	2	0.00467	0.009	0.011
65^2	3	4	0.014	0.028	0.030
129^2	6	15	0.07	0.098	0.107
257^2	25	50	0.22	0.37	0.39
513^2	300	700	1.15	1.87	1.69
1025^2	5400	12400	6.5	9.1	9.6
2049^2	86800	189300	-	39.8	41.3
4097^2	more than 668000	-	-	182	190

given in Table 2.3. The corresponding CPU-time scaling in log-log for different methods are compared in Fig. 2.14 (d). The cost of computations (in terms of CPU-time) of direct and multigrid methods are proportional to the number of grid points (N^2 for two-dimensions), but for iterative methods this is increasing exponentially $\text{CPU}_{\text{time}} = 5 \exp(0.01 N)$, which is very restrictive for computations on fine grids. Some comments are addressed as follows:

1. By optimizing the FFT the proposed direct method can be more efficient. The resolution of the finest possible grid on the available machine is 4096^2 .
2. The memory limitation of multigrid solver developed by Paknejad [133] is restrictive on fine grids, the finest possible resolution is 1024^2 on the available machine.
3. From parallelization view point the multigrid solver is the most difficult but the iterative methods are the easiest to parallelize. In Sachs et al. [137] a parallel implementation of global multigrid method via implicit partitioned method is presented.
4. The proposed direct method can be parallelized by the pencil rotation method as is done in [131] for a direct method using operator splitting in alternative directions. Nearly linear strong scaling (speed up) is reported by Laizet and Lamballais in [131].

2.4 Time integration

Because of high accuracy and straightforward parallelization, the explicit fourth-order Runge-Kutta method is one of the best and mostly used methods for integration of ordinary differential equations. By collecting all discretized spatial derivatives in the *RHS* operator one can solve the considered partial differential equation $\partial_t \omega = \text{RHS}(\partial_{\mathbf{x}}(), \partial_{\mathbf{xx}}(), \dots)$ as a system of ordinary differential equation at each time step. This RK4 method is a four

stage method including two predictor and two corrector steps, see Fig. 2.16.

First step ($i = 1$):

$$\begin{aligned} k_1 &= RHS(\omega, \psi)^n \\ \omega_i &= \omega^n + \frac{\Delta t}{2} k_1 \\ -\nabla^2 \psi_i &= \omega_i \end{aligned}$$

Second step ($i = 2$):

$$\begin{aligned} k_2 &= RHS(\omega, \psi)^{i-1} \\ \omega_i &= \omega^n + \frac{\Delta t}{2} k_2 \\ -\nabla^2 \psi_i &= \omega_i \end{aligned}$$

Third step ($i = 3$):

$$\begin{aligned} k_3 &= RHS(\omega, \psi)^{i-1} \\ \omega_i &= \omega^n + \Delta t k_3 \\ -\nabla^2 \psi_i &= \omega_i \end{aligned}$$

Fourth step ($i = 4$):

$$\begin{aligned} k_4 &= RHS(\omega, \psi)^{i-1} \\ \omega^{n+1} &= \omega^n + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4) \\ -\nabla^2 \psi^{n+1} &= \omega^{n+1} \end{aligned} \tag{2.87}$$

For integration of vorticity transport equation (2.5) we have

$$RHS(\omega, \psi)^i = (-\partial_y \psi \partial_x \omega + \partial_x \psi \partial_y \omega + \nu \nabla^2 \omega + \partial_x F_y - \partial_y F_x)^i \tag{2.88}$$

where $i = 1, 2, 3, 4$. At each time step, Eq. (2.88) must be evaluated four times, in which Eq. (2.6) must be solved to update the stream-function ($-\nabla^2 \psi_i = \omega_i$). For details and technical discussions of the Runge–Kutta methods we refer to [56]. In addition to one memory location for u , five memory locations are necessary at each grid point for the evaluation of k_1, k_2, k_3, k_4 and u^* . However, Δt is limited by CFL (Courant–Friedrich–Levy) condition, which implies that

$$\Delta t \leq CFL \frac{\Delta x}{U} \tag{2.89}$$

where U is an advection (or a phase) velocity. In the case of nonlinear advection in space more attention must be paid. In the presence of viscous (dissipation) term also a viscous

constraint of the form

$$\Delta t \leq VSL \frac{\Delta x^2}{\nu} \quad (2.90)$$

must be respected. For integration via an explicit method we have

$$CFL = \frac{\sigma_i}{w'_{\max}} \approx 1 \quad , \quad VSL = \frac{\sigma_r}{w''_{\max}} \approx 0.4 \quad (2.91)$$

where $[-\sigma_r, 0]$ and $[-i\sigma_i, +i\sigma_i]$ are real and imaginary limits of the stable region of the time integration methods in the complex plane. We refer to Fig. 2.15 and Table 2.5 for real and imaginary limits of the stable region of different time integration methods. As can be seen for the Runge–Kutta family in contrast to the Adams–Bashforth methods, by increasing the precision, the stable region becomes larger. However, the evaluation of spatial derivatives in the Adams–Bashforth methods is performed only once per time-step, in contrast to the Runge–Kutta methods where the order of the method determines the number of evaluations of the spatial derivatives. In general the memory usage by the Runge–Kutta methods with the same order of accuracy is more than the Adams–Bashforth methods. However low storage Runge–Kutta methods can be advantageous in terms of memory allocation. Here w'_{\max} and w''_{\max} represent respectively the maximum values of the scaled modified wavenumbers for the first and second derivatives, corresponding to Eqs. (2.39) and (2.40) which are plotted in Fig. 2.6 (a) and Fig. 2.8 (a). The values of w'_{\max} can be approximately extracted from Fig. 2.6 (a) for different methods to avoid the explicit calculation of the derivative of Eq. (2.39). As can be seen, we have $w'_{\max} \in [1, \pi]$ for the first derivative. For the second-derivative the maximums are located at $w = \pi$. Thus by replacing $\omega = \pi$ in equation (2.40) we have

$$w''_{\max} = w''(\pi) = \frac{4a + 4c/9}{1 - 2\alpha + 2\beta} \quad (2.92)$$

according to the Eq. (2.92), we have $w''_{\max} \in [4, \pi^2]$ for the second derivative. The values of w'_{\max} and w''_{\max} for different central differentiation schemes are listed in Table 2.4. Stability limits of some explicit time integration methods, via linear analysis, are listed in Table 2.5 and are also shown in Fig. 2.15. For viscous flows (low Reynolds number) the time-step is more restricted by VSL constraint than by CFL . Therefore an implicit integration of viscous terms is preferable specially when the grid is stretched. In the presence of moving bodies the displacement of the moving object must not exceed the grid spacing, i.e.,

$$\Delta t \leq \frac{\Delta \mathbf{x}}{\mathbf{u}_B}$$

With the use of the explicit penalization method the following constraint for Δt , must also be respected.

$$\Delta t \leq \eta$$

Table 2.4: Maximum values of modified wavenumbers for the first and second derivatives via central finite difference methods.

method	w'_{max}	w''_{max}
2^{nd} -order explicit	1.000	4.000
4^{th} -order explicit	1.372	5.333
6^{th} -order explicit	1.584	6.044
4^{th} -order 3D-implicit	1.735	6.000
6^{th} -order 3D-implicit	1.988	6.857
8^{th} -order 3D-implicit	2.128	7.324
8^{th} -order 5D-implicit	2.205	7.471
10^{th} -order 5D-implicit	2.324	7.838
Spectral-like 5D-implicit	2.632	9.108
Spectral (exact)	$3.14(\pi)$	$9.86(\pi^2)$

Table 2.5: Stability limits of some explicit methods via linear analysis [63].

method	σ_i	σ_r
Adams–Bashforth 1	0	2
Adams–Bashforth 2	0	1
Adams–Bashforth 3	0.73	0.56
Adams–Bashforth 4	0.43	0.3
Euler (RK1)	0	2
Runge–Kutta 2	0	2
Runge–Kutta 3	1.7	2.5
Runge–Kutta 4	2.8	2.8

Finally an error analysis of the time integration for the viscous Burgers equation

$$\partial_t u + uu_x = \nu u_{xx}$$

is performed to examine the accuracy of the time integration based on the fourth-order Runge–Kutta method. The aim is to show the rate of convergence of different errors with successive reduction of the time step. A simulation with $\Delta t_{max}/16$ is considered as reference solution to compute the errors, where Δt_{max} is chosen rather large, e.g., equal to 0.1, to avoid the truncation error from falling in the range of the round-off error for $\Delta t_{max}/16$. On the other hand the spatial resolution was chosen sufficiently fine, i.e., $N = 2048$, to avoid the intervention of truncation errors due to the second-order discretization of the spatial terms. Other simulations were performed with $\Delta t_{max}/8$, $\Delta t_{max}/4$, $\Delta t_{max}/2$ and Δt_{max} ,

successively. The computations start from an initial condition

$$u(x, 0) = \sin(x), \quad x \in [0, 2\pi]$$

at $t = 0$ and stop at $t = 1$, so the time step and the number of iterations for each simulation is different. Homogeneous Dirichlet boundary conditions are applied at $x = 0$ and $x = 2\pi$. Different errors as a function of $\Delta t_{max}/\Delta t$ are compared with the theoretical slope of -4 in Fig. 2.17. A good agreement can be observed. The developed solver will also be used for time integration of the vorticity transport equation (2.5) in two dimensional problems. In the cases where the first-order Euler method is used for time integration, this will be mentioned explicitly.

2.5 The algorithm for fluid–structure interaction

The summary of the algorithm for fluid–structure interaction is listed in Algorithm 1 (where $\alpha_1 = 1/2$, $\alpha_2 = 1/2$ and $\alpha_3 = 1$). The flowchart is illustrated in Fig. 2.18.

2.6 Conclusion

In this Chapter an algorithm for the simulation of moving bodies interacting with two dimensional incompressible flows was proposed. By using a uniform Cartesian grid a new fourth-order direct solver for the solution of the Poisson equation is presented which combines a compact finite difference with a FFT in alternative directions. The advantages of our method are fourth-order accuracy, convergence down to machine zero over an optimal grid, compact tridiagonal stencils, possibility of extension to three dimensions, reduced arithmetics and memory usage in comparison to iterative methods. Moreover, the parallelization is straightforward because decoupling of the operations in different directions is done. Nearly linear strong scaling (speed up) and efficiency is reported by Laizet and Lamballais [131] for a similar direct solver by decoupling of the operators in alternative directions. The efficiency and accuracy of the solver are compared with an iterative and a multigrid method. For introducing a moving body in fluid flow, the volume penalization method is applied to the solution of the Navier–Stokes equations as a forcing term. Fourth-order compact finite difference discretization of the curl of volume penalization terms is shown to be equivalent to the diffused-interface immersed body method proposed by Lai and Peskin [76]. An advantage of the volume penalization method is that the evaluation of the hydrodynamic coefficients is straightforward. Proper denoising of the hydrodynamic coefficients is crucial in dealing with fluid–solid interaction problems via the volume penalization method. Extensions and validation of the proposed algorithm for a variety of fluid–solid interaction problems, will be presented in the following Chapters.

Algorithm 1 Fluid–structure interaction

1. START FROM AN INITIAL CONDITION
 2. BODY KINEMATICS
 - (a) Compute the mask $\chi(i, j)$ and smooth it by Eq. (2.52)
 - (b) Compute the moment of inertia J around reference point
 - (c) Compute body velocities $u_p(i, j), v_p(i, j)$ in Eulerian grid (Lagrange \rightarrow Euler)
 3. TIME INTEGRATION OF FLOW FIELD VIA RK4
 - (a) $\omega_0 = \omega^n$
 - (b) $\psi_0 = \psi^n$
For $i = 1, 2, 3$ ($\alpha_1 = 1/2, \alpha_2 = 1/2$ and $\alpha_3 = 1$)
 - (c) Compute $k_i(\omega, \psi)^{i-1}$ from Eq. (2.88)
 - (d) $\omega_i = \omega^n + \alpha_i \Delta t k_i$
 - (e) Solve Eq. (2.6); $-\nabla^2 \psi_i = \omega_i$
End For
 - (f) Compute $k_4(\omega, \psi)^3$ from Eq. (2.88)
 - (g) Update vorticity from Eq. (2.87); $\omega^{n+1} = \omega^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
 - (h) Solve Eq. (2.6); $-\nabla^2 \psi^{n+1} = \omega^{n+1}$
 4. SOLVE FOR THE BODY DYNAMICS
 - (a) Compute the hydrodynamic coefficients from Eqs. (2.22) and (2.23)
 - (b) Denoise the coefficients by Eq. (2.25)
 - (c) Compute displacements from Eq. (2.28)
 - (d) Compute rotation from Eq. (2.30)
 5. WRITE NECESSARY DATA TO FILE
 6. IF $T < T_{end}$, GO TO STEP 2
 7. END
-

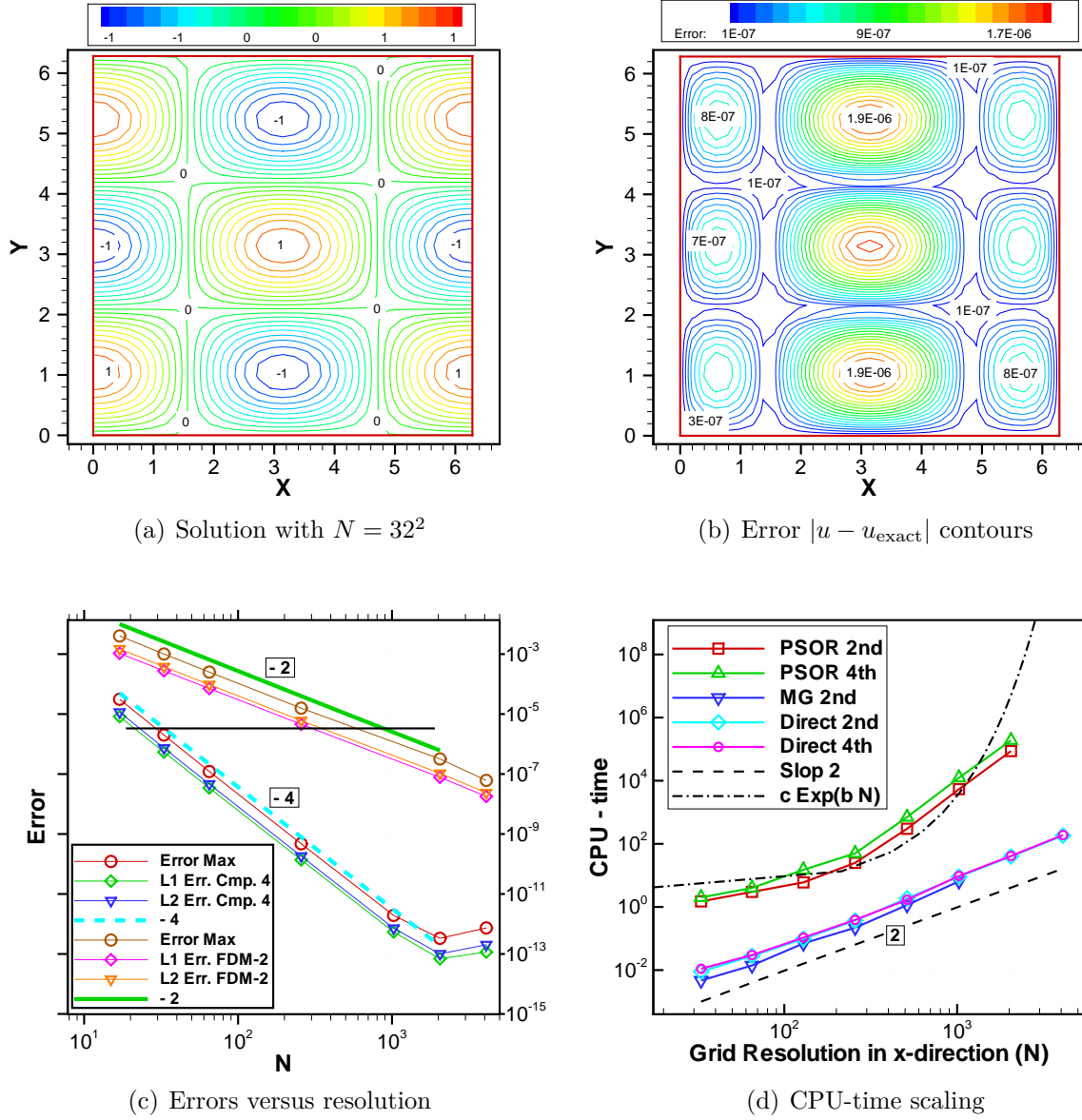


Figure 2.14: (a) Solution of the Poisson equation ($\nabla^2 u = f$) with a fourth-order compact method, forcing term is $f(x, y) = -(n^2 \pi^2 / L_y^2 + 1) \cos(x) \sin(n \pi y / L_y)$, $(x, y) \in \Omega = [0, 2\pi]^2$ and Dirichlet boundary conditions are given by $u_b(x, y) = \cos(x) \sin(n \pi y / L_y)$, (x, y) on $\partial\Omega$, ($n = 3$). (b) The corresponding error contours $E(x, y) = |u(x, y) - u_{\text{exact}}(x, y)|$, $(x, y) \in \Omega$ in comparison with the exact solution. (c) Error analysis for direct Poisson solvers computed with second and fourth order compact methods. (d) CPU-time scaling of different iterative (Multigrid / Point Successive Over Relaxation) and direct methods.

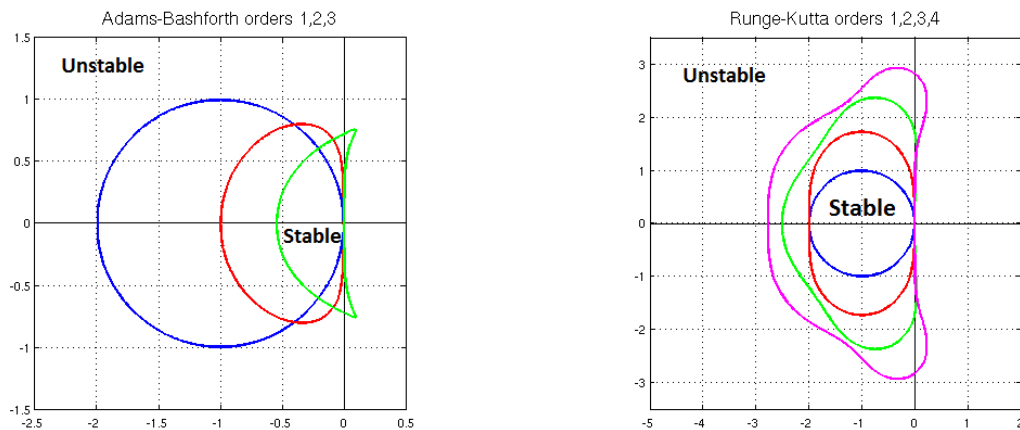


Figure 2.15: Stable regions for time integration via Adams–Bashforth and Runge–Kutta methods on complex plan. The picture is taken from [63].

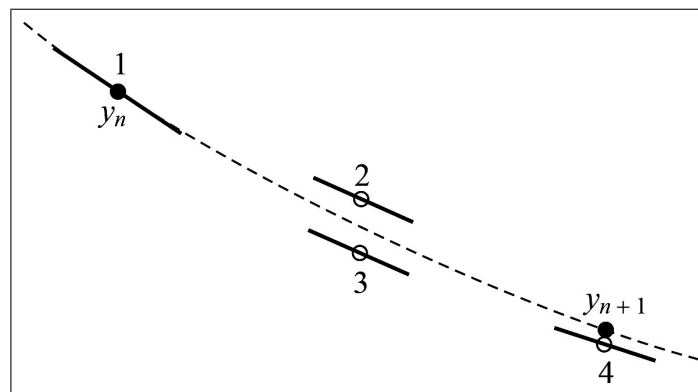


Figure 2.16: Schematic representation of the fourth-order Runge–Kutta method. In each time step the *RHS* operator must be evaluated four times: once at the initial point (marked \bullet), twice at trial midpoints (marked \circ) and once at a trial endpoint (marked \circ). From these derivatives the value of the function in the next time step (marked \bullet) can be calculated. The picture is taken from [56].

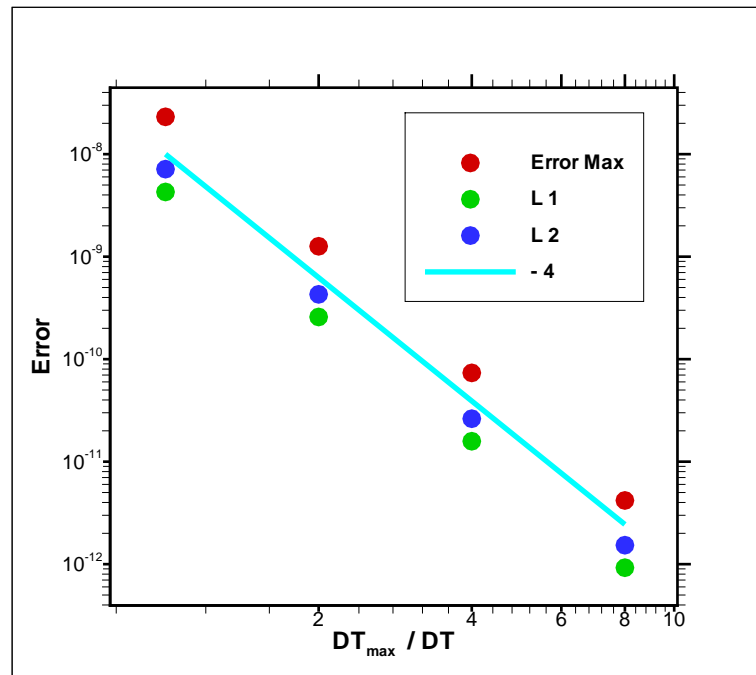


Figure 2.17: Time accuracy of explicit fourth-order Runge–Kutta method applied to the Burgers equation over uniform grid with $\nu = 10^{-3}$.

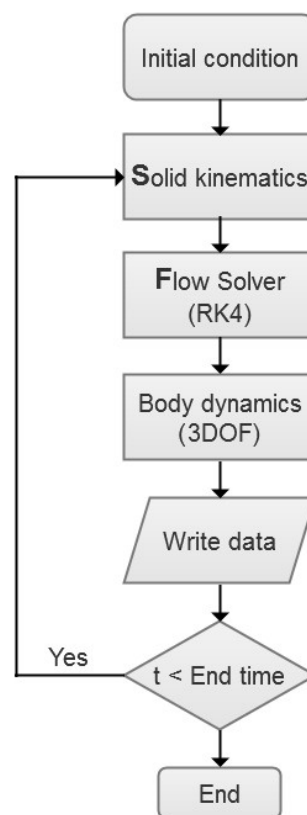


Figure 2.18: Flowchart of the fluid–solid interaction (FSI) algorithm.

Chapter 3

Convergence study and validation of the proposed algorithm

“It seems the surge of progress began immediately after the war has now largely spent itself we have got down to the bedrock difficulty of solving non-linear PDEs [96].”

G. K. Batchlor (1953)

In this chapter, first numerical simulation of decaying Taylor vortices, which is an exact solution of the Navier–Stokes equations, will be presented. The spatial errors of the developed solver without the penalization term will be assessed. Then for examination of the error introduced by the penalization term, the Taylor–Couette flow will be secondly considered. Next the unsteady flow around a circular cylinder is considered for evaluation of the hydrodynamic forces via two different methods. Finally, for validation of the fluid–structure interaction, the free fall of a cylinder and an ellipse in a quiescent fluid due to terrestrial gravity will be studied.

3.1 Decaying Taylor vortices

Taylor vortices are an analytical solution of the Navier–Stokes equations without forcing term ($F = 0$) represented in Cartesian coordinates. The boundary conditions are Dirichlet but are time varying. Following Chorin [22], Kim and Moin [44] the solution is given in primitive variables

$$u(\mathbf{x}, t) = -\cos x \sin y e^{-2\nu t} \quad (3.1)$$

$$v(\mathbf{x}, t) = \sin x \cos y e^{-2\nu t} \quad (3.2)$$

$$p(\mathbf{x}, t) = -\rho/2(\cos^2 x + \cos^2 y) e^{-4\nu t} + \text{cte} \quad (3.3)$$

The pressure can also be represented as follows

$$p(\mathbf{x}, t) = -\rho/4(\cos 2x + \cos 2y) e^{-4\nu t} + \text{cte} \quad (3.4)$$

By considering the Navier–Stokes equations (2.1) one can see that, the local acceleration is equal to the viscous dissipation and the convective terms are in balance with the pressure gradient. This vortical flow includes stagnation points, with zero velocity and maximum pressure. The points with minimum pressure are located in the center of the vortices. In non-dimensional form ν can be replaced by Re^{-1} . The Poisson equation in the form of

$$\nabla^2 p = \rho(u_x v_y - u_y v_x)$$

is satisfied with vanishing normal pressure gradients (homogeneous Neumann) $\partial p / \partial n = 0$ at the boundaries $\partial\Omega$ of a square domain ($0 \leq x \ \& \ y \leq 2\pi$). For vorticity and stream-function formulation we have

$$\omega(\mathbf{x}, t) = 2 \cos x \cos y e^{-2\nu t} \quad (3.5)$$

$$\psi(\mathbf{x}, t) = \cos x \cos y e^{-2\nu t} + \text{cte} \quad (3.6)$$

where the advection terms cancel each other in the vorticity transport equation (2.5) and the viscous terms represent the time-decay. Poisson equations in the form of $-\nabla^2 \psi = \omega$ or

$$\nabla^2 p = 2\rho(\psi_{xx}\psi_{yy} - \psi_{xy}^2)$$

are also satisfied with proper boundary conditions. An arbitrary domain of solution can be considered, e.g., a circle or a diamond, by setting proper boundary conditions for each variable according to the pre-cited Eqs (3.1)-(3.6). At $t = 0$ a divergence free initial condition is achieved with the given relations for the velocity components. In this section the spatial accuracy of the developed solvers are examined via an error analysis. The methods used for discretization of the spatial terms are, an explicit second order method and a fourth order compact method (classical Padé). Following Abide and Viazzi [97] we use decaying Taylor vortices with $\nu = 0.001$ for showing second and fourth order convergence of the spatial error. The analytical solution is given by Eqs (3.1)-(3.6). The contours of the vorticity and stream-function are shown in Fig. 3.1 (a) and (b) respectively, at the end of the simulation i.e., after 1000 iterations. The time step is choosed in the order of machine zero to minimize the error introduced by time integration for all of the spatial resolutions. In Fig. 3.1 (c) and (d) different errors for the vorticity and stream-function versus grid resolution, via second and fourth order methods, are shown respectively. The slopes of different errors versus grid resolution are in agreement with analytical -2 and -4 slopes. As expected, in both of the analyses a saturation of the error by round-off error on

fine grids can be seen.

3.2 Taylor–Couette flow

Taylor–Couette flow [5] consists of a viscous fluid (with kinematic viscosity ν) confined between two concentric cylinders with radii (R_1, R_2) in rotation with different angular velocities (Ω_1, Ω_2) , see Fig. 3.2. The Taylor number,

$$Ta = \frac{R_1(\Omega_2 - \Omega_1)^2(R_2 - R_1)^3}{\nu^2}$$

characterizes the importance of centrifugal or inertial forces due to rotation, relative to viscous forces. For Ta below the critical value $Ta_c \approx 1708$, the flow is steady and purely azimuthal ($u_z = u_r = 0$). This state is known as circular Taylor–Couette flow and for which we have an analytical solution which is independent of viscosity. For reason of convenience the solution is represented in cylindrical coordinates. The azimuthal velocity is given by

$$u_\theta(r) = Ar + \frac{B}{r} \quad , \quad (r, \theta) \in [R_1, R_2] \times [0, 2\pi] \quad (3.7)$$

where

$$A = \frac{\Omega_2 R_2^2 - \Omega_1 R_1^2}{R_2^2 - R_1^2} \quad , \quad B = \frac{R_1^2 R_2^2 (\Omega_1 - \Omega_2)}{R_2^2 - R_1^2}$$

The pressure is given by

$$p(r) = \frac{A^2}{2} r^2 + 2AB \ln(r) - \frac{B^2}{2} r^{-2} \quad (3.8)$$

The vorticity between two cylinders is constant ($\omega_z = 2A$). The stream-function is given by

$$\psi(r) = -\frac{A}{2} r^2 - B \ln(r) + c_0 \quad (3.9)$$

where c_0 must be determined with respect to an arbitrary reference point. If one uses the volume penalization method, the velocity components must be enforced in solid regions from known angular velocities (i.e., Ω_1 and Ω_2),

$$u_\theta(r) = r\Omega \quad , \quad (r, \theta) \in [0, R_1] \cup [R_2, R_{max}] \times [0, 2\pi] \quad (3.10)$$

The vorticity inside the rotating regions is constant and is equal to twice of the domain angular velocity ($\omega_z = 2\Omega$) and the stream-function is given by

$$\psi(r) = -\frac{\Omega}{2} r^2 + c \quad , \quad (r, \theta) \in [0, R_1] \cup [R_2, R_{max}] \times [0, 2\pi] \quad (3.11)$$

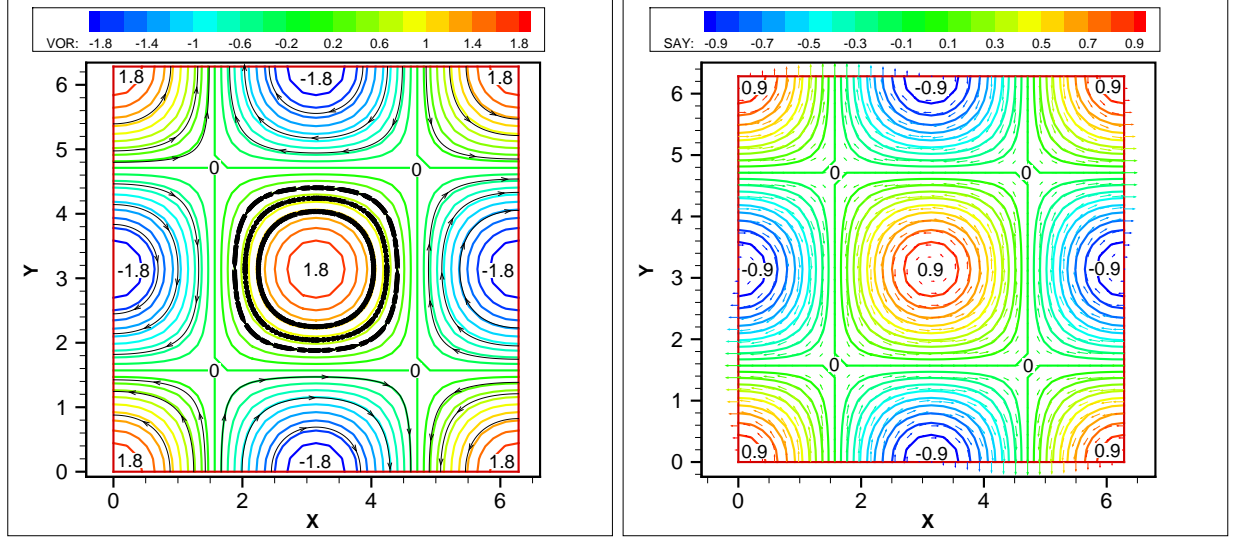
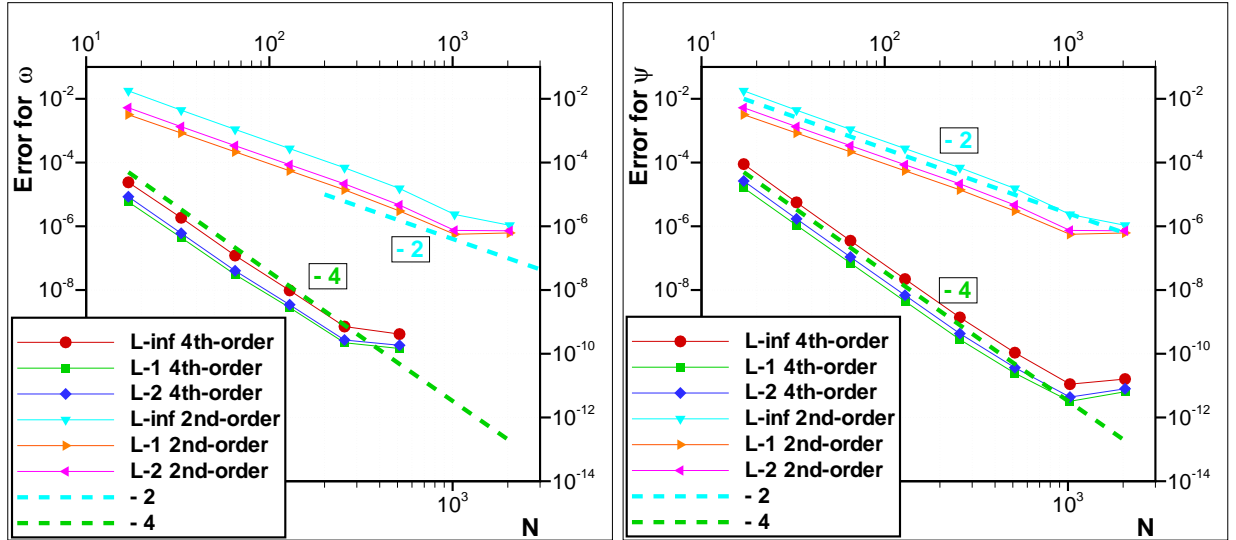
(a) Contours of ω (b) Contours of ψ (c) Errors of ω versus grid resolution(d) Errors of ψ versus grid resolution

Figure 3.1: Vorticity (a) and stream-function (b) contours for Taylor vortices, $(x, y) \in [0, 2\pi] \times [0, 2\pi]$ at $t = 0.0001$. Error analysis for vorticity (c) and stream-function (d), computed with second and fourth order finite differences.

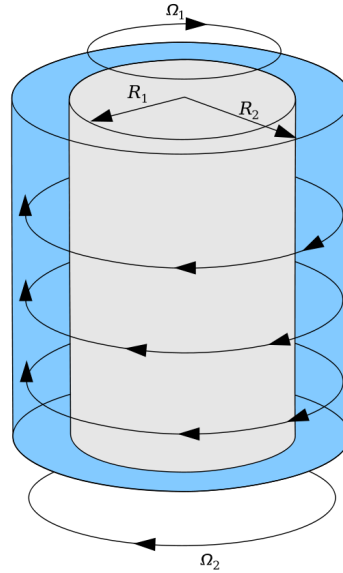


Figure 3.2: Setup of a Taylor–Couette flow, picture from Wikipedia.

where c must be determined for each domain in accordance with c_0 . In Cartesian coordinates we have $(u, v) = (-u_\theta \sin \theta, u_\theta \cos \theta)$. For a rigorous study of the error due to the volume penalization term added to the Navier–Stokes equations in vorticity-stream function formulation, an exact solution is necessary. Taylor–Couette configuration is a good choice, first and foremost, because of the known Dirichlet boundary conditions everywhere, and secondly, because of the presence of curved walls contrary to other analytical solutions usually represented in Cartesian domains which can coincide with the underlying Cartesian grid used to discretize the governing equations. That is to say although the solver is adapted to a Cartesian domain, in this case the mask function which is the representative of penalized area is curved (see Fig. 3.3). This configuration is similar to practical test cases, like the flow around circular cylinder or complex geometries which will be considered in the following. A second-order finite difference method is used for discretization of the governing equations including the curl of the penalization term $\nabla \times \mathbf{F}$. The L_1 -error $\|u^{\text{exact}} - u_\eta^{\text{num}}\|$ for u , which is the x component of the velocity field, is calculated for different penalization parameters η and resolutions (N in x and y directions). The simulations are carried out until a steady state is reached, so that the error is independent of the time discretization. A unit square domain is considered as the solution domain, the time-step is calculated by the constraints presented in the Section 2.4 and the kinematic viscosity is fixed to $\nu = 0.01$. The radii are chosen as $R_1 = 0.2$ and $R_2 = 0.4$, respectively. At $t = 0$ the fluid domain is at rest and the inner-cylinder is set into movement with a fixed angular velocity ($\Omega_1 = 0.2$) while the angular velocity of the outer cylinder is kept equal to zero ($\Omega_2 = 0$). The Taylor number for this configuration ($Ta = 0.64$) is below the critical value, thus the flow is purely azimuthal. The computations are stopped when the time $t_{\text{end}} = 10$ is reached. At this instant, the changes in errors are negligible (invisible), which

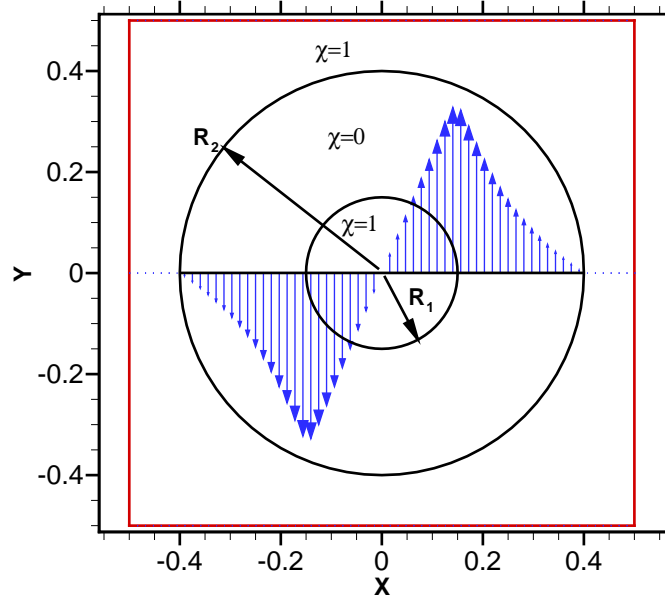


Figure 3.3: Schematic representation of a penalized unit square domain for modeling of Taylor–Couette flow with volume penalization method ($\chi = 0$ represents the fluid domain and $\chi = 1$ the solid domains respectively). The radius of the inner cylinder is $R_1 = 0.15$ and that of the outer cylinder is $R_2 = 0.4$. The angular velocity of the inner cylinder is $\Omega_1 = 0.2$ and that of the outer is equal to zero, $\nu = 0.01$ and $Ta \approx 1$.

indicates that a steady state has been reached. The mask is filtered (mollified), with Eq. (2.52) presented in Section 2.2.4. Original and mollified Mask functions at the midline, i.e., $y = 0.5$, are illustrated in Fig. 3.4 (a). A comparison of the computed vorticity ω , the stream-function ψ and the velocity u with exact solutions, using $N = 128$ grid point in each direction, is plotted in Fig. 3.4 (a)-(b). The convergence of the L_1 -error of u versus the grid resolution, for different penalization parameters are shown in Fig. 3.4 (c), where between first and second order convergence can be seen. Suppose u_η^{num} denotes the numerical solution of the penalized equation, for quantifying the numerical error of u_η^{num} compared to u^{exact} (the solution to the original Navier-Stokes problem), the error can be estimated by

$$\|u^{\text{exact}} - u_\eta^{\text{num}}\| \leq \underbrace{\|u^{\text{exact}} - u_\eta\|}_{O(\sqrt{\eta})} + \underbrace{\|u_\eta - u_\eta^{\text{num}}\|}_{O(\Delta x^p)} \quad (3.12)$$

where $\|\cdot\|$ is an appropriate norm. The first term at the right-hand side is the error due to the penalization term and the second term represents the discretization error (p being the formal order of accuracy of the numerical method used to discretize the equation). According to Nguyen et al. [164] a compromise between two errors is to chose $\Delta x \approx \sqrt{\eta}$, which leads to a first-order convergence for the error $\|u^{\text{exact}} - u_\eta^{\text{num}}\| \leq O(\Delta x)$. The convergence of the L_1 -error of u versus different penalization parameters is shown for

different grid resolution in Fig. 3.4 (d), which is shown to be of order $\sqrt{\eta}$. For these calculations the expected formal accuracy $p = 2$ is found and the convergence is between first and second order in space as a function of the resolution N , confirming the analysis of Carbou and Fabrie [92] and Morales et al. [163]. We also observe a saturation of the convergence error for large N , corresponding to domination of the penalization error. An optimal resolution can be found for each η . Decreasing η leads to accuracy enhancement in general but for an explicit time integration Δt is also limited by η as discussed in Section 2.4. A fine grid will also need a small η as can be seen in the Fig. 3.4.

3.3 Unsteady flow around a circular cylinder

For verification of the developed solver, an incompressible flow over a circular cylinder at $Re = 200$ is considered, which is one of the most thoroughly investigated unsteady flows. A short review of the involved methods is recalled in the following. For better temporal resolution a fourth-order Rung-Kutta method is used for time integration of the vorticity and stream-function version of the Navier–Stokes equations in two-dimensions. The non-dimensional physical time-step was set according to the constraints explained in Section 2.4. A Cartesian uniform grid in both directions is used on a rectangular domain. The volume penalization method is used to introduce the cylinder in the solution domain. The choice of η is prescribed by the error analysis done for the Taylor–Couette flow in Section 3.2. Central second-order finite differences method is used for discretization of all spatial derivatives. Two multigrid codes developed by Paknejad [133] and by Mousavinia [161] are used for accelerating the rate of convergence of the elliptic part of the algorithm. The elliptic equation is the most time consuming part of the calculations and guarantees the incompressibility constraint. A Full Multigrid (FMG) method is used in [133] and [161]. By using “V” cycles the solution starts on the coarsest grid, then advances by interpolation toward the finer grids. The method uses the Gauss-Seidel iteration with red–black sweeper as the *smoothing operator*. For the *prolongation operator* bilinear interpolation is employed. Then half-weighting is used for the *restriction operator*. A grid independent solution must be achieved, thus the resolution of the grid must be fine enough to be able to capture the main physical phenomena like frequency of vortex shedding (Strouhal number) or some integral quantities like hydrodynamic coefficients. However, due to the between first and second order accuracy of the volume penalization method realization of a perfect boundary layer near the solid surface seems to be unattainable. Nevertheless the interests of this method in terms of efficiency and applicability for moving and deformable bodies, which is the main subject of the present study, motivate and justify the use of the volume penalization method. Alongside the fact that an external flow is considered for the first time with the present method, a validation of the equations (2.21)–(2.23) for calculation of the hydrodynamic forces and torque using the volume penalization method will be done in

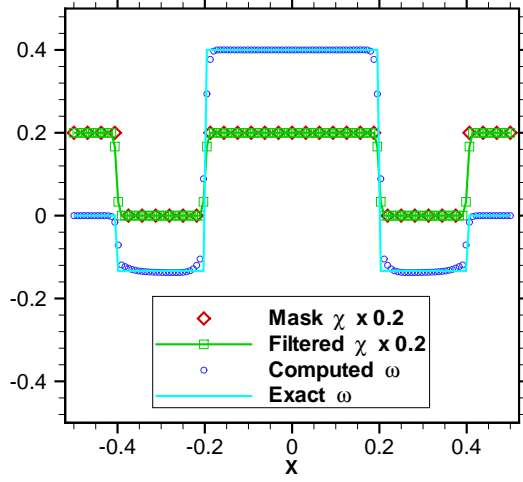
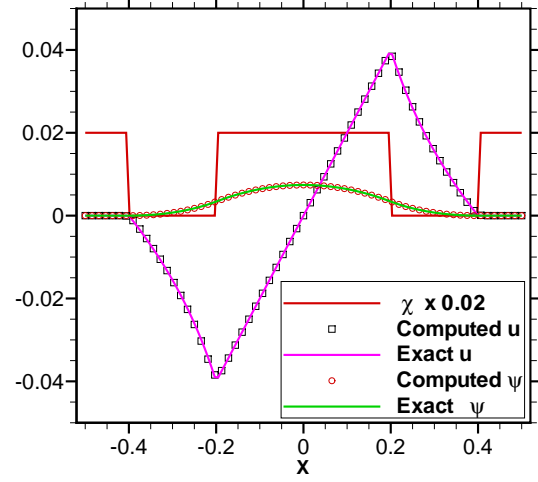
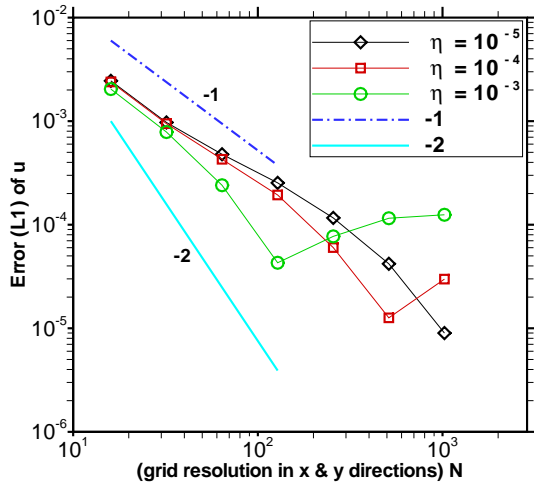
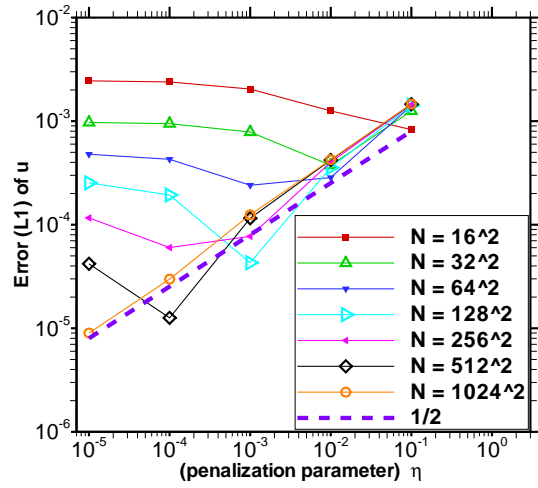
(a) Couette flow, cut at $y = 0.5$ (b) Couette flow, cut at $y = 0.5$ (c) Error of u versus resolution(d) Error of u versus η

Figure 3.4: (a) Original and mollified mask function, comparison of computed vorticity ω with the exact solution, using $N = 128$ grid points in each direction. (b) Comparison of the computed stream-function ψ and velocity u with the exact solution, using $N = 128$. (c) The L_1 -error of u with the spatial resolution (N being the grid resolution in each direction). (d) The L_1 -error of u versus the penalization parameter η .

Table 3.1: Boundary conditions for ψ and ω on a rectangular domain $\Omega \in [0, L_x] \times [0, L_y]$ which is used for the simulation of the flow around circular cylinder.

B. C.	Right (outflow)	Left (inflow)	Up (inviscid wall)	Down (inviscid wall)
ψ	$\partial_x \psi = 0$	$\partial_y \psi = U_\infty$	$\psi = c_2$ (no-penetration)	$\psi = c_1$ (no-penetration)
ω	$\partial_t \omega = -u \partial_x \omega$	$\omega = 0$	$\omega = 0$ (free-slip)	$\omega = 0$ (free-slip)

the following using this test case. The simulations were started from stationary uniform flow conditions ($\omega = 0$, $\partial_y \psi = U_\infty$) and were continued until periodic shedding of vortices occurred. Boundary conditions of the rectangular domain at the inflow is uniform flow ($u = U_\infty$, $v = 0$). At the outflow $v = 0$ is imposed to obtain ψ ,

$$\psi_N = (4\psi_{N-1} - \psi_{N-2})/3$$

then an advection condition for vorticity is considered. More complicated or simpler options for boundary conditions at the outflow can be considered. However the results of the presented boundary conditions are satisfactory. All boundary conditions are summarized in Table 3.1.

3.3.1 Hydrodynamic forces via a control volume around body

Usually the hydrodynamic forces exerted by the fluid onto the body are calculated by surface integrals $\mathbf{F} = \oint_{\partial\Omega_p} \boldsymbol{\sigma} \cdot \mathbf{n} dS$ of the stress tensor $\sigma_{i,j}(\mathbf{u}, p) = -p\delta_{i,j} + \mu S_{i,j}$. With the use of immersed boundary methods the flow variables are usually not available at the surface of the body and must be interpolated. As a consequence the calculated forces may be not accurate. By using the volume penalization method, volume integration is already presented in Section 2.1.1 to determine the forces. We present an alternative to that method and then compare the results. Here we consider a time dependent rectangular penetrable domain Ω_{cv} (control volume) around the immersed body Ω_p (see Fig. 3.5). The Leibnitz-Reynolds transport theorem gives us the time derivative of the integral of a time dependent variable (e.g. linear momentum) over a time dependent domain,

$$\frac{d}{dt} \int_{system(t)} \rho \mathbf{u}(t) dV = \frac{\partial}{\partial t} \int_{cv} \rho \mathbf{u} dV + \oint_{cs} \rho \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) dS \quad (3.13)$$

For a closed system the second law of Newton is written as $\mathbf{F} = d(m\mathbf{u})/dt$. Therefore the hydrodynamic forces exerted onto the body can be deduced from integral forms of the

Table 3.2: Comparison of the parameters/methods used for simulation of the unsteady flow around circular cylinder at $Re = 200$.

Simulation	$\Delta x = L_x/n_x$	$\Delta y = L_y/n_y$	Δt	η	Initial/inflow Noise	Force evaluation via
2	24/2049	12/1025	2×10^{-3}	2×10^{-3}	0/0	VP
3	24/1025	12/513	2×10^{-3}	2×10^{-3}	0/0	VP
4	24/4097	12/2049	1×10^{-3}	1×10^{-3}	0/0	VP
5	24/1025	12/513	1×10^{-3}	1×10^{-3}	$10^{-1}/10^{-4}$	CV/VP
6	24/513	12/257	2×10^{-3}	2×10^{-3}	$10^{-3}/10^{-4}$	CV/VP

Navier–Stokes equations. Following Davidson [96] for the linear momentum we have:

$$\frac{\partial}{\partial t} \int_{cv} \rho u_i dV = - \oint_{cs} u_i (\rho \mathbf{u} \cdot \mathbf{n}) dS + \underbrace{\oint_{cs} \sigma \cdot \mathbf{n} dS}_{\text{Surface forces}} + \underbrace{+ \mathbf{F}_i^V}_{\text{Volume forces}} \quad (3.14)$$

where $i = 1, 2, 3$ and the angular momentum is given by:

$$\frac{\partial}{\partial t} \int_{cv} \rho (\mathbf{r} \times \mathbf{u})_i dV = - \oint_{cs} (\mathbf{r} \times \mathbf{u})_i (\rho \mathbf{u} \cdot \mathbf{n}) dS + \oint_{cs} \mathbf{r} \times (\sigma \cdot \mathbf{n}) dS + \mathbf{M}_i^V \quad (3.15)$$

Thereby the forces can be evaluated by the following relation [145]:

$$\mathbf{F} = \underbrace{- \frac{d}{dt} \int_{cv} \rho \mathbf{u} dV}_{\text{Momentum variation rate}} + \underbrace{\oint_{cs} \sigma \cdot \mathbf{n} dS}_{\text{Exerted forces by fluid}} - \underbrace{\oint_{cs} \rho (\mathbf{u} - \mathbf{u}_p)(\mathbf{u} \cdot \mathbf{n}) dS}_{\text{Momentum net flux from cs}} - \underbrace{\oint_{\partial\Omega_p} \rho (\mathbf{u} - \mathbf{u}_p)(\mathbf{u} \cdot \mathbf{n}) dS}_{\text{Momentum flux from body surface}} \quad (3.16)$$

If no-slip boundary conditions are imposed on the body, the last term is equal to zero and can be eliminated. The moments acting on the body (by the control volume) can be derived by vector product of the forces with the distance vector $\mathbf{r} = \mathbf{x} - \mathbf{x}_{ref}$ from the reference point as follows,

$$\mathbf{M} = - \frac{d}{dt} \int_{cv} \rho \mathbf{r} \times \mathbf{u} dV + \oint_{cs} \rho \mathbf{r} \times (\sigma \cdot \mathbf{n}) dS - \oint_{cs} \rho \mathbf{r} \times (\mathbf{u} - \mathbf{u}_p)(\mathbf{u} \cdot \mathbf{n}) dS \quad (3.17)$$

where \mathbf{n} is the outward-pointing unit normal vector to the control surface (cs) and \mathbf{u}_p is the velocity of the immersed body which is considered equal to the velocity of the surrounding control volume (cv). For validation of the presented methods some simulations of the unsteady flow around a circular cylinder at Reynolds number equal to 200 (known as von Kármán vortex shedding) are performed. The parameters used in the different computations are given in Table 3.2. Hydrodynamic coefficients, Strouhal number and

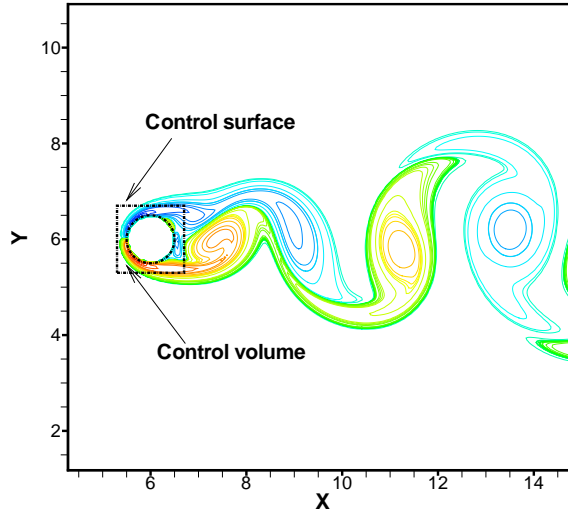


Figure 3.5: Sketch of the domain $(\Delta X, \Delta Y) \in]1.2, 1.2[$ used to compute the hydrodynamic coefficients via control volume method (equivalent to control surface and surrounding lines in two dimensions).

transition time are compared with the results presented in the literature in Table 3.3. The lift coefficient is defined as,

$$C_l = \frac{F_L}{\frac{1}{2}\rho_f U_\infty A_{ref}}$$

the drag coefficient as,

$$C_d = \frac{F_D}{\frac{1}{2}\rho_f U_\infty A_{ref}}$$

and the moment coefficient as,

$$C_m = \frac{M_z}{\frac{1}{2}\rho_f U_\infty A_{ref} L_{ref}}$$

where F_L and F_D are normal and parallel to the flow forces represented in [Newton], M_z is the pitching moment in $[N.m]$, $q = \frac{1}{2}\rho_f U_\infty$ is the dynamic pressure, $A_{ref} = D \times 1$ is the reference area and $L_{ref} = D$ is the reference length. The Strouhal number is a dimensionless frequency of vortex shedding, $St = f \frac{D}{V}$, determined from the time variation of lift. The transition time is also obtained from the curve of lift, i.e., it is measured between the last instance of observing a complete symmetric wake, corresponding to the first instability in the flow, up to the fully developed periodic vortex shedding state. The results obtained in the present study, are compared in Table 3.3 with numerical simulations performed by other researchers as well as with experimental measurements. Fig. 3.6 shows the snapshots of vorticity contours obtained during a simulation, started from a motionless initial condition up to a developed von Kármán shedding. One can see in (a)-(c) that vorticity sheets start to

Table 3.3: Comparison of the results obtained from the developed code for simulation of the unsteady flow around a circular cylinder at $Re = 200$ with those of other researchers. Comparison is done for Strouhal number (determined from the time variation of lift), hydrodynamic coefficients (lift, drag and moment) and the transition time over the curve of the lift coefficient.

Reference	St	C_l	C_d	C_m	Transition start-end
Present study					
Simulation 3 VP	0.193	± 0.84	1.58 ± 0.062	± 0.23	112-210
Simulation 2 VP	0.198	± 0.78	1.49 ± 0.054	± 0.054	105-202
Simulation 4 VP	0.201	± 0.72	1.42 ± 0.056	± 0.025	30-145
Simulation 5 VP	0.19	± 0.8	1.58 ± 0.07	± 0.44	1-37
Simulation 6 CV	0.19	± 0.82	1.64 ± 0.06	-	10-50
Simulation 6 VP	0.19	± 0.86	1.7 ± 0.07	± 0.98	10-50
Valizadeh et al. [124]					
Upwind 3 rd -order	0.182	± 0.75	1.32 ± 0.05	-	-
Upwind 5 th -order	0.192	± 0.68	1.33 ± 0.045	-	25-75
Tai & Zhao [93]	0.195	± 0.64	1.31 ± 0.041	-	5-57
Rogers et al. [48]					
Upwind 3 rd -order	0.160	± 0.75	1.29 ± 0.05	-	-
Upwind 5 th -order	0.185	± 0.65	1.23 ± 0.05	-	24-115
Lecoite & Piquet [42]					
Compact 2 nd -order	0.227	± 0.7	1.46 ± 0.04	-	-
Compact 4 th -order	0.194	± 0.5	1.58 ± 0.0035	-	-
Rosenfeld et al. [46]	0.211	± 0.69	1.46 ± 0.05	-	-
Linnick & Fasel [98]					
$\lambda = 0.056$	0.199	± 0.70	1.37 ± 0.046	-	-
$\lambda = 0.023$	0.197	± 0.69	1.34 ± 0.044	-	-
Liu et al. [68]	0.192	± 0.69	1.31 ± 0.049	-	-
Wang & Zhang [149]	0.198	± 0.69	1.32 ± 0.04	-	-
Belov et al. [60]	0.193	± 0.64	1.19 ± 0.042	-	-
Miyake et al. [58]	0.196	± 0.67	1.34 ± 0.043	-	-
Taira et al. [117]	0.196	± 0.68	1.35	-	-
Martinez [37]	-	-	1.27 ± 0.0035	-	-
Lin et al. [34]	-	-	1.17	-	-
Thoman & Szewczyk [24]	-	-	1.17 ± 0.005	-	-
Wille [18] (experimental)	-	-	1.3	-	-
Roshko [12] (exp.)	0.19	-	-	-	-
Kovaszny [10] (exp.)	0.19	-	-	-	-
Berger & Wille [31] (exp.)	0.18-0.19	-	-	-	-

develop with opposite signs from up and down of cylinder, then prolonged to downstream in a symmetric manner until the amplification of small perturbations (which are present in nature and numerical simulations). The presence of shear in the mean flow (in the wake region) provides an appropriate ambiance for instability growth. The instability causes distortion of the plane sheets of vorticity, to perform a wavy motion. Then because of incompressibility constraint the wavy sheets are divided to rotational structures. Due to stability considerations, the rotational structures are more stable in comparison to sheet-like vorticity. Finally, vortices with opposite signs shed from up and down of the cylinder known as Kármán vortex shedding. In the present study the maximum stable separation length is about $L_{sep} = 10D$. Generally it depends on the numerical method, the perturbation of the inflow and also to the initial condition. Fig. 3.7 shows the computed lift, drag and moment on the cylinder versus non-dimensional time obtained with the present solver. Lift and moment start from zero, pass through a transitional oscillatory state and evolve to a perfect sinusoidal oscillation. In the work of Valizadeh [124] the drag curve shows a minimum of about $C_d = 0.9$ at $t = 35$ before the start of the oscillation.

Considering the CPU-time necessary to obtain a fully developed periodic vortex shedding, initial perturbations and noise in the inflow can help to bypass the transition state. Initial perturbations are defined as: $u = U_\infty + 0.01 \times u'$, $v = 0.01 \times v'$ and noise in the inflow are defined as: $u = U_\infty + 10^{-4} \times u'$, $v = 10^{-4} \times v'$, where $u', v' \in [0, 1]$ are random numbers. The results of the simulation with perturbations at the inflow and initial condition are demonstrated in Fig. 3.8. The corresponding hydrodynamic forces are given in Fig. 3.10. Comparison of the hydrodynamic forces and the torque, calculated via the surrounding control volume and the volume penalization method is illustrated in Fig. 3.10 (a). Components (pressure, momentum, shear and volume) of drag and lift forces calculated by control volume method are also given in Fig. 3.10 (b) and (c), respectively.

3.4 Fluid–structure interaction

3.4.1 Free sedimentation of a cylinder

In this section we perform a simulation of a two-dimensional cylinder, falling due to the gravity in a quiescent fluid, to validate the two-way fluid/solid interaction. We compare our results with those of Gazzola et al. [144] and Namkoong et al. [119] which have the same physical parameters. A rigid 2D cylinder of diameter $D = 0.005$ m with $\rho_b = 1.01\rho_f$, is released from rest in a fluid with density $\rho_f = 996$ kg/m³ and kinematic viscosity $\nu = 8 \times 10^{-7}$ m²/s and accelerates due to gravity ($g = -9.81$ m/s²) until it reaches its asymptotic terminal velocity. The domain size is set to $(x, y) \in [0, 0.04 \text{ m}] \times [0, 0.32 \text{ m}] = [0, 8D] \times [0, 64D]$. The spatial resolutions in our simulations are set to 512×4096 and

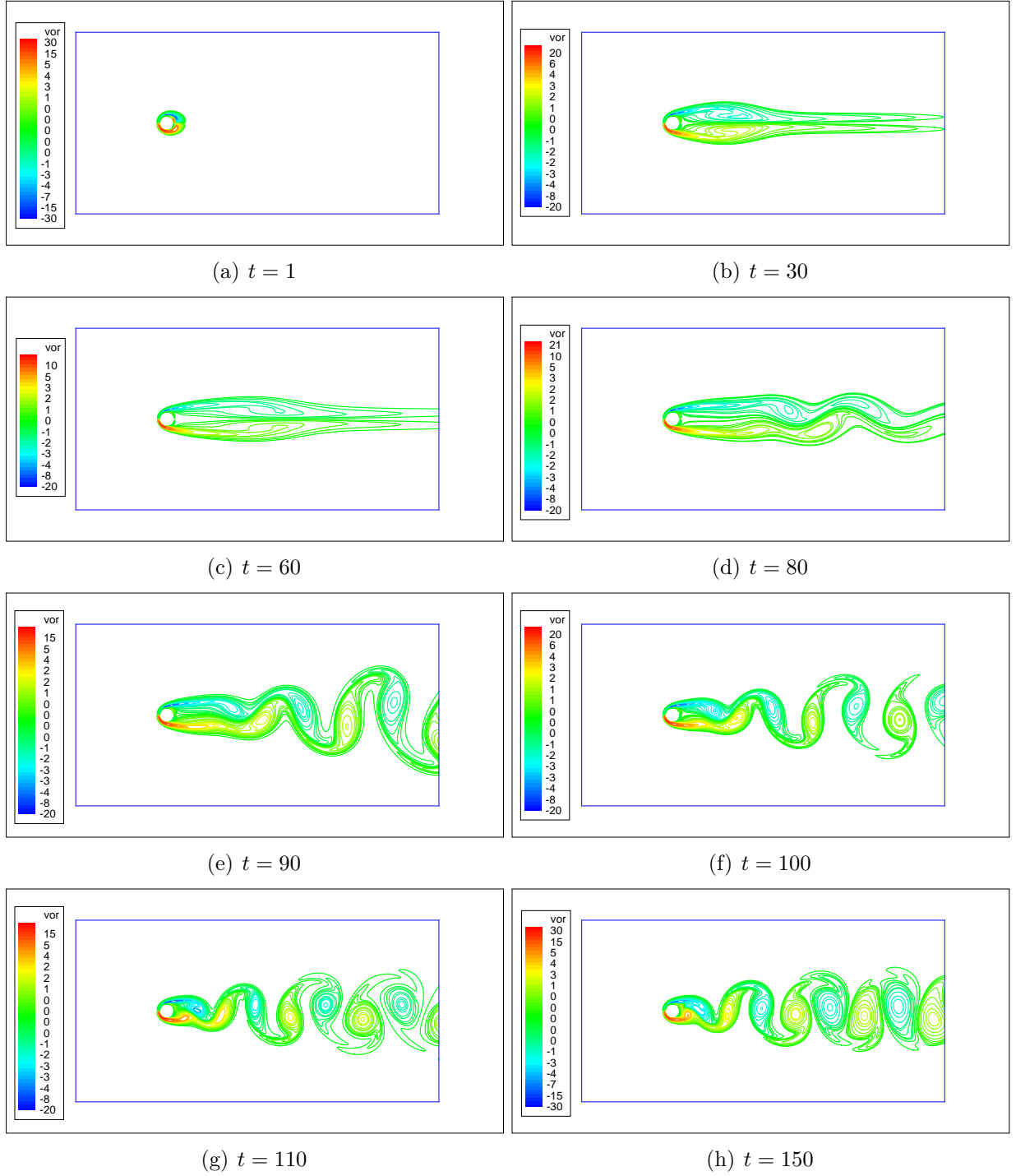


Figure 3.6: Vorticity contours around a circular cylinder (simulation 4), where $Re=200$, $\Delta x = \frac{24}{4097}$, $\Delta y = \frac{12}{2049}$, $x_{cg} = L_x/4$, $y_{cg} = L_y/2$, $\Delta t = 10^{-3}$ and $\eta = 10^{-3}$.

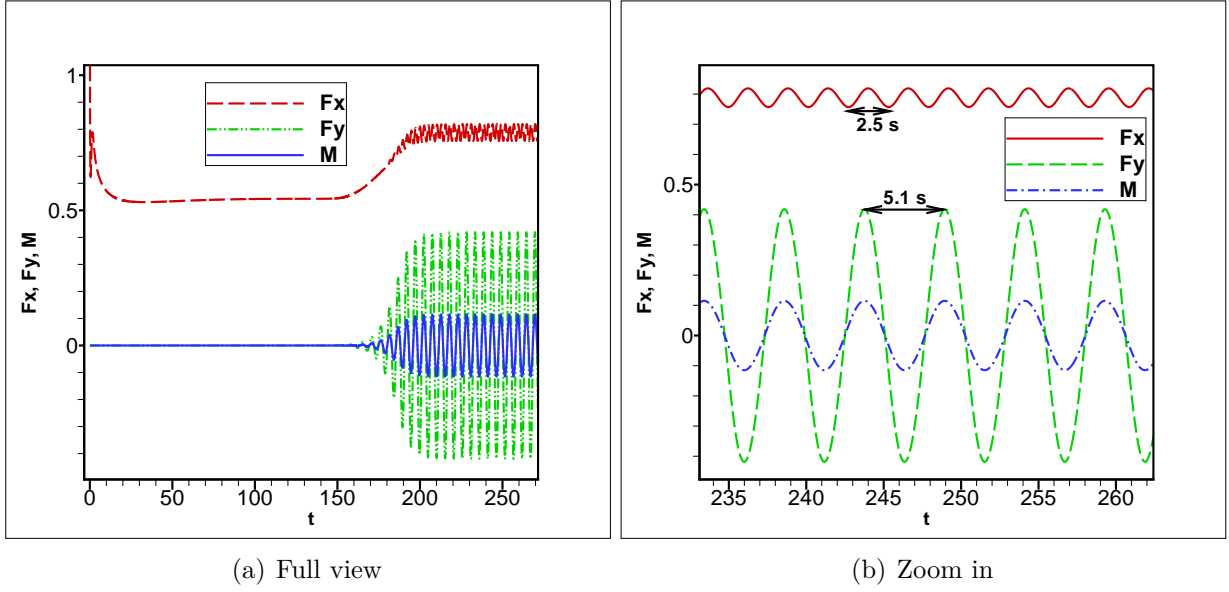


Figure 3.7: Hydrodynamic forces and moment for the circular cylinder (simulation 3), where $Re=200$, $\Delta x = \frac{24}{1025}$, $\Delta y = \frac{12}{513}$, $\Delta t = 2 \times 10^{-3}$ and $\eta = 2 \times 10^{-3}$.

1024×8192 , the penalization parameter $\eta \in [10^{-4}, 10^{-3}]$, the time step $\Delta t \in [10^{-4}, 10^{-3}]$ and the filter parameter for denoising the hydrodynamic coefficients is $\delta \in [0.001, 0.005]$. Second and fourth order discretizations are used in the simulations. In the simulations of Gazzola et al. [144] the resolution is 1024×8192 , the penalization parameter $\eta = 10^{-4}$ and the Lagrangian CFL is set to 0.01.

The snapshots of the vorticity isolines generated by the falling cylinder in a fully quiescent and slightly perturbed fluid are illustrated in Fig. 3.12 and Fig. 3.13, respectively. An example of the initial perturbation, created with a hyperbolic tangent function, for triggering the transition during the sedimentation of a cylinder is illustrated in Fig. 3.11. A qualitative agreement with the simulations of Gazzola et al. [144] can be observed. Comparison of the vortical structures at $t = 13$ between the simulation with slightly perturbed initial condition ($\mathbf{u}' \approx 0.001 \times \text{random number} \times u_\infty$) represented in Fig. 3.13 (h) and that of a fully quiescent initial condition represented in Fig. 3.12 (h), shows that the presence of perturbations in the initial condition can trigger the transition in the early stage of the fall, i.e., $t \approx 3$. This is particularly important to obtain comparable results with other simulations with different numerical methods where the added numerical dissipation is not necessarily the same. Without adding any initial perturbations the transition can be triggered (e.g., at $t \approx 10$) by the numerical errors which are performing like a perturbation (see Fig. 3.12). This kind of transition is not controlled, it depends on the grid resolution and the numerical implementation and explains the delayed streamwise velocity overshoot and the different transient flow fields.

Fig. 3.14 shows the time evolution of the streamwise and lateral velocities obtained

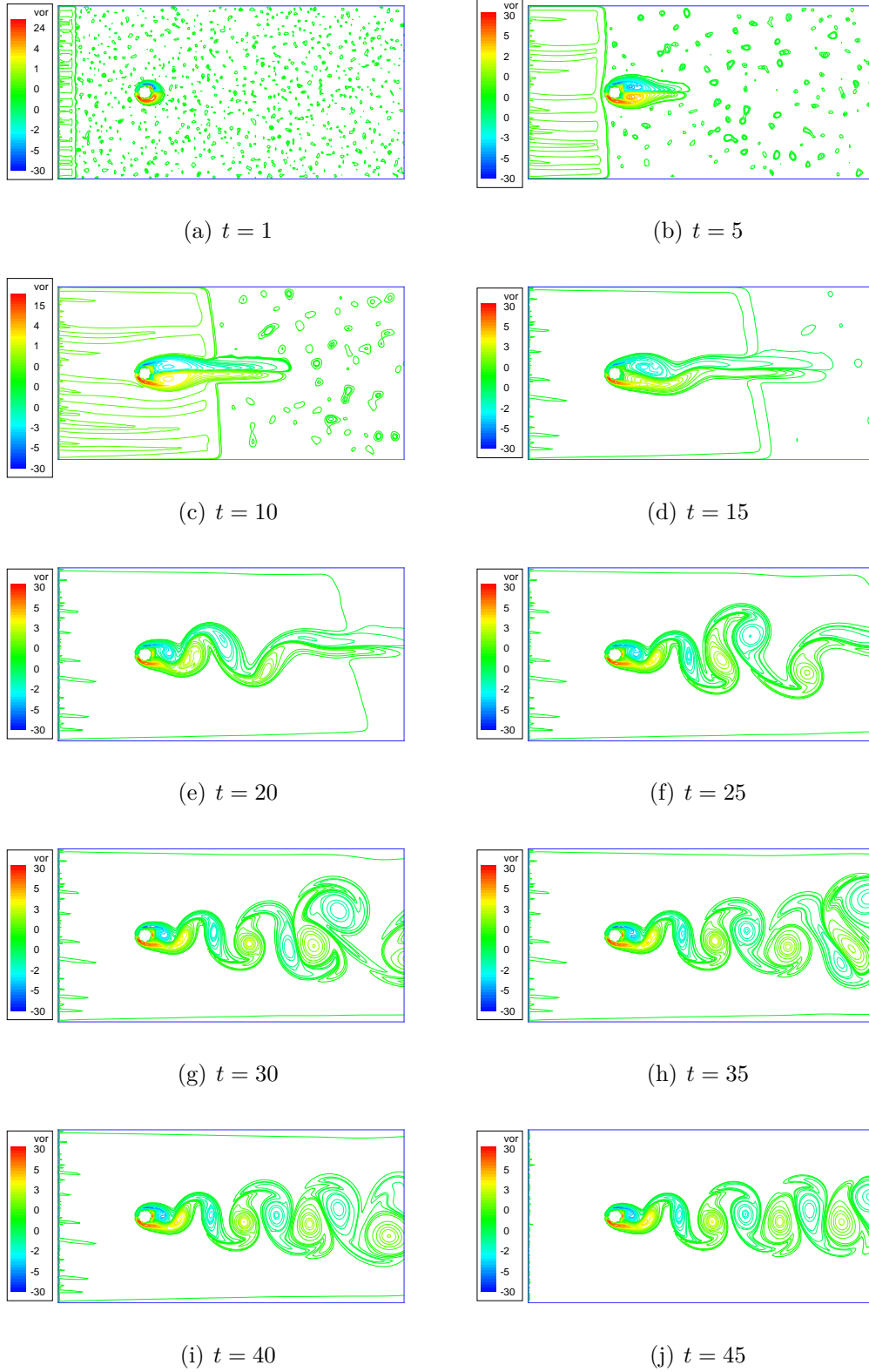


Figure 3.8: Vorticity contours of the flow around a circular cylinder (simulation 5) started by an initial perturbation ($u = U_\infty + 0.01 \times u'$, $v = 0.01 \times v'$) and noise in the inflow ($u = U_\infty + 10^{-4} \times u'$, $v = 10^{-4} \times v'$) where $u', v' \in [0, 1]$ are random numbers, $Re=200$, $\Delta x = \frac{24}{1025}$, $\Delta y = \frac{12}{513}$, $x_{cg} = L_x/4$, $y_{cg} = L_y/2$, $\Delta t = 10^{-3}$ and $\eta = 10^{-3}$.

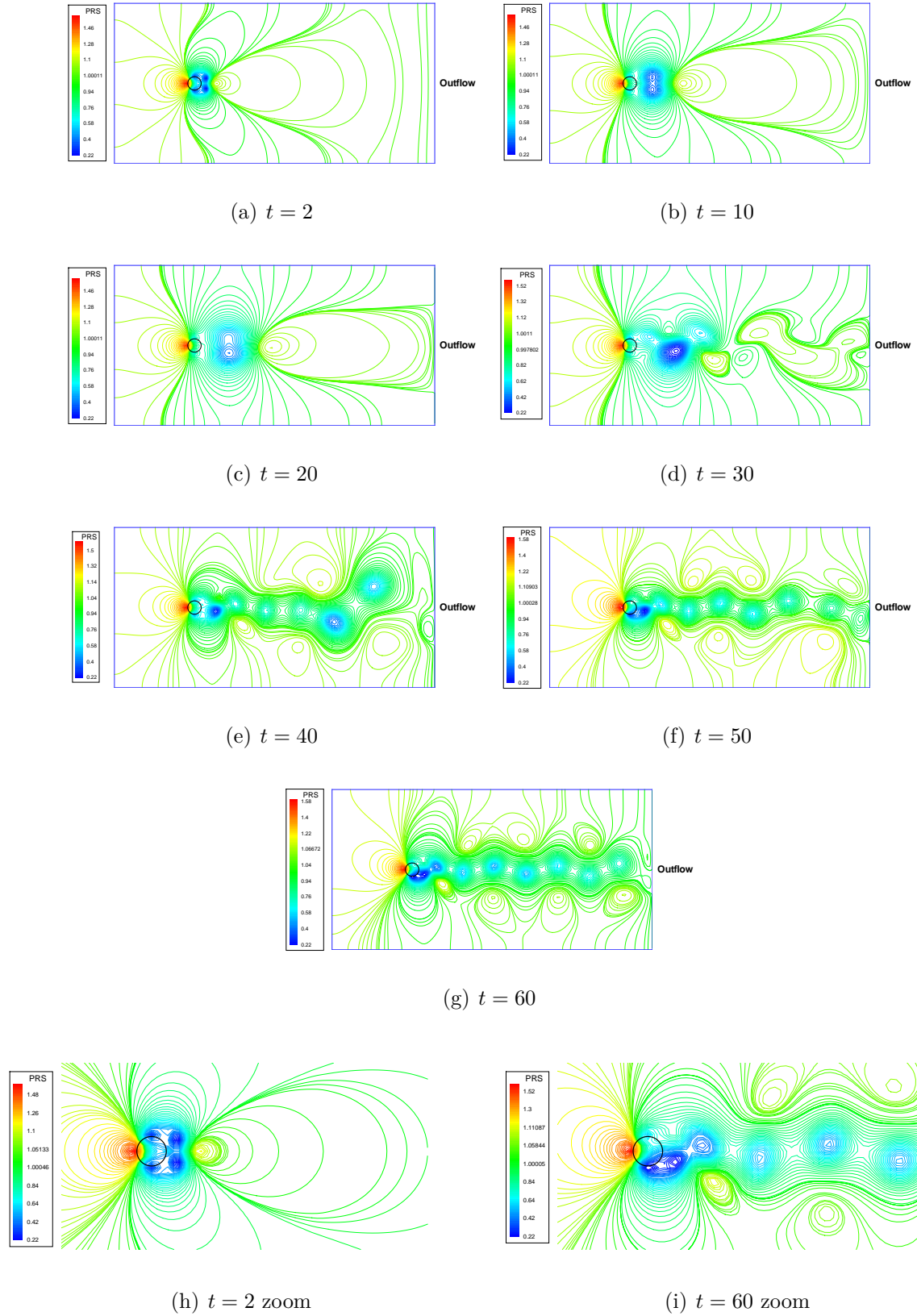


Figure 3.9: Pressure contours of the flow around a circular cylinder (simulation 6) started by an initial perturbation ($u = U_\infty + 10^{-3} \times u'$, $v = 10^{-3} \times v'$) and noise in inflow ($u = U_\infty + 10^{-4} \times u'$, $v = 10^{-4} \times v'$) where $u', v' \in [0, 1]$ are random numbers, $Re=200$, $\Delta x = \frac{24}{513}$, $\Delta y = \frac{12}{257}$, $x_{cg} = L_x/4$, $y_{cg} = L_y/2$, $\Delta t = 2 \times 10^{-3}$ and $\eta = 2 \times 10^{-3}$.

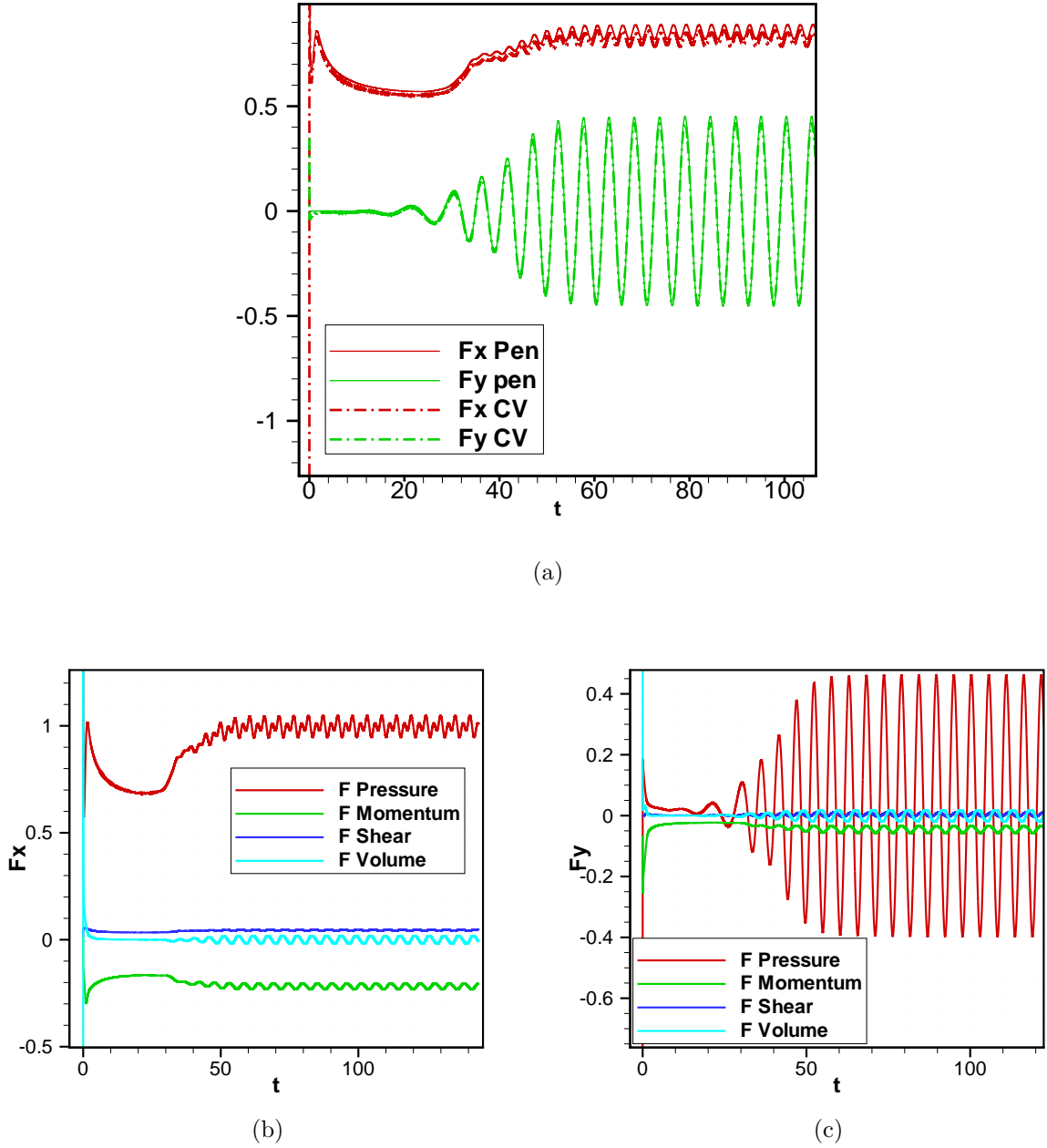


Figure 3.10: (a) Comparison of the hydrodynamic forces of the circular cylinder, calculated via the surrounding control volume and the volume penalization method (simulation 6), started by an initial perturbation ($u = U_\infty + 10^{-3} \times u'$, $v = 10^{-3} \times v'$) and noise in inflow ($u = U_\infty + 10^{-4} \times u'$, $v = 10^{-4} \times v'$) where $u', v' \in [0, 1]$ are random numbers, $Re=200$, $\Delta x = \frac{24}{513}$, $\Delta y = \frac{12}{257}$, $\Delta t = 2 \times 10^{-3}$ and $\eta = 2 \times 10^{-3}$. (b) Components (pressure, momentum, volume and shear) of the drag force $F_x = -F_x P - F_x M - F_x V + F_x S$ calculated by the CV method. (c) Components (pressure, momentum, volume and shear) of the lift force $F_y = -F_y P - F_y M - F_y V + F_y S$ calculated by the CV method.

with the present method and those of Namkoong et al. [119] and Gazzola et al. [144]. As can be seen the streamwise velocity shows the same dynamics as the reference simulations. In particular the streamwise velocity obtained in the simulation with perturbed initial condition (see Fig. 3.11) overshoots above the terminal velocity and then slows down when the vortices start shedding. It can be seen that because of 8D width of the domain size, there is no significant difference between the streamwise velocities by imposing no-slip and free-slip boundary conditions at the boundaries of the rectangular Cartesian grid. After transition the amplitude of the oscillations of the lateral velocity $u_{\text{lateral}} = \pm 0.002$ m/s is in agreement with those of Namkoong et al. [119] and Gazzola et al. [144], but a phase shift can be observed due to a short delay in the transition in our simulation. The terminal streamwise velocity of the simulation with perturbed initial condition is $u_{\text{streamwise}} = 0.024$ m/s which corresponds to Reynolds number $Re \approx 150$ and that of the unperturbed initial condition is $u_{\text{streamwise}} = 0.025$ m/s which corresponds to Reynolds number $Re \approx 156$. In the former an overshoot can be observed in the streamwise velocity while in the later the overshoot takes place in a larger time interval or it is entirely eliminated. The terminal velocity differs less than 5% from the reference terminal velocity in the case of the perturbed initial condition and coincides in the case of the unperturbed initial condition. The differences are due to different Poisson solvers which is unbounded in the simulation of Gazzola et al. [144], the boundary conditions which are free-slip and no-penetration in our simulations, different penalization parameters and resolutions. From the authors viewpoint the take-home message here is that the near one relative solid/fluid density leads to a small buoyancy where an invalid approximation of the hydrodynamic coefficients especially in the early stages of the fall yields the simulation to a failure. To cope with this challenge the process of denoising of the hydrodynamic coefficients with a proper filter parameter δ_{filter} is devised in the proposed algorithm to eliminate the non physical oscillations of the hydrodynamic coefficients.

3.4.2 Validation of the solid dynamics with a falling ellipse

For further validation of the proposed algorithm to deal with rotating objects interacting with incompressible flows, sedimentation of an ellipse due to terrestrial gravity field is considered in this section. According to Kolomenskiy and Schneider [140] different behaviors like steady falling, fluttering, tumbling and chaotic motion can be observed by varying the ellipse aspect ratio a/b , density ratio ρ_b/ρ_f and the viscosity ν of the fluid. These parameters can be summarized in a dimensionless moment of inertia

$$J_{\text{cg}}^* = 2J_{\text{cg}}/(\pi a^4 \rho_f) = (a^2 + b^2)(b/2a^3)(\rho_b/\rho_f)$$

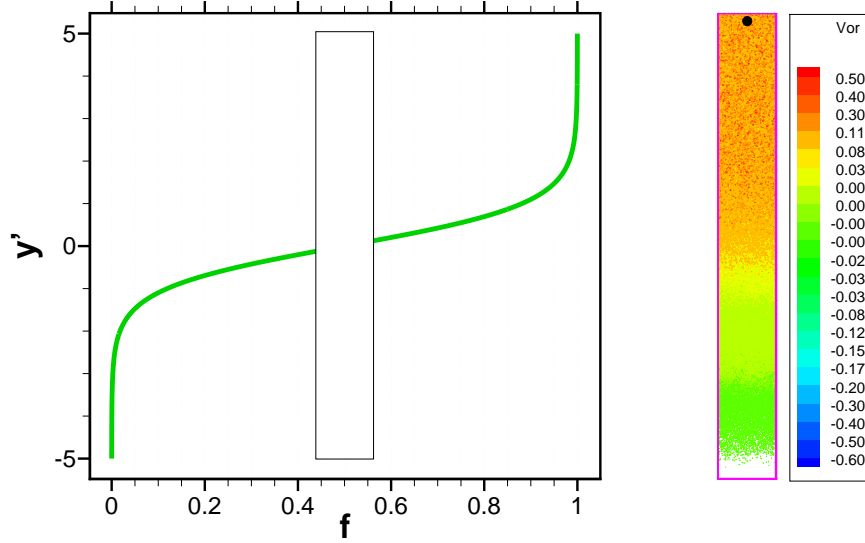


Figure 3.11: An example of the initial perturbation created with a hyperbolic tangent function for triggering the transition during the sedimentation of a cylinder $\mathbf{u}(\mathbf{x}) = f \times \text{noise} \times \mathbf{u}'$, where $f = 0.5(\tanh y' + 1)$, $\text{noise} = 10^{-3} \times U_{ref}$, $\mathbf{u}' \in [-1, 1]$ are uniformly distributed random numbers, $U_{ref} = \mathbf{u}_{max}$, $y' = 10y/L_y - 5$, $y' \in [-5, 5]$, $y \in [0, 1]$ and vorticity $\omega = v_x - u_y$.

and the Reynolds number $Re = u_t L / \nu$, where u_t is the sedimentation average velocity estimated by Gazzola et al. [144] with

$$u_t = \sqrt{4bg(\rho_b/\rho_f - 1)} \quad (3.18)$$

Kolomenskiy and Schneider [126] have replaced the coefficient 4 by π in the definition of the reference velocity Eq. (3.18). In our opinion the definition of the reference velocity by Eq. (3.18) is questionable and needs more investigation. Using Eq. (3.18) for evaluation of the reference velocity leads to under prediction of the Reynolds number. We think that the average velocity $u_t = (\bar{U}_{cg}^2 + \bar{V}_{cg}^2)^{1/2}$ in the final stage of the fall would be a better choice. Nevertheless instead of dimensionless numbers, we use the ellipse aspect ratio, density ratio and the viscosity of the fluid as influencing parameters, to classify the behavior of the ellipse in fall. The results of the three simulations performed by the second order solver for the falling ellipse corresponding to steady fall, fluttering and tumbling are reported in the following. The domain of the solution for steady fall and fluttering is $(x, y) \in [0, 5L] \times [0, 20L]$ where $L = 2a = 1$ and $H = 2b = 0.2$ are the major and minor diameters of the ellipse, respectively. The resolution of the grid is $N_x \times N_y = 512 \times 2048$. For simulation of falling ellipse in the tumbling regime a larger domain and a finer grid are needed. Therefore $(x, y) \in [0, 10] \times [0, 10]$ and $N_x \times N_y = 2048 \times 2048$ are used for the simulation of the tumbling regime. Decreasing the kinematic viscosity from $[\text{m}^2/\text{s}]$, $\nu = 0.03$

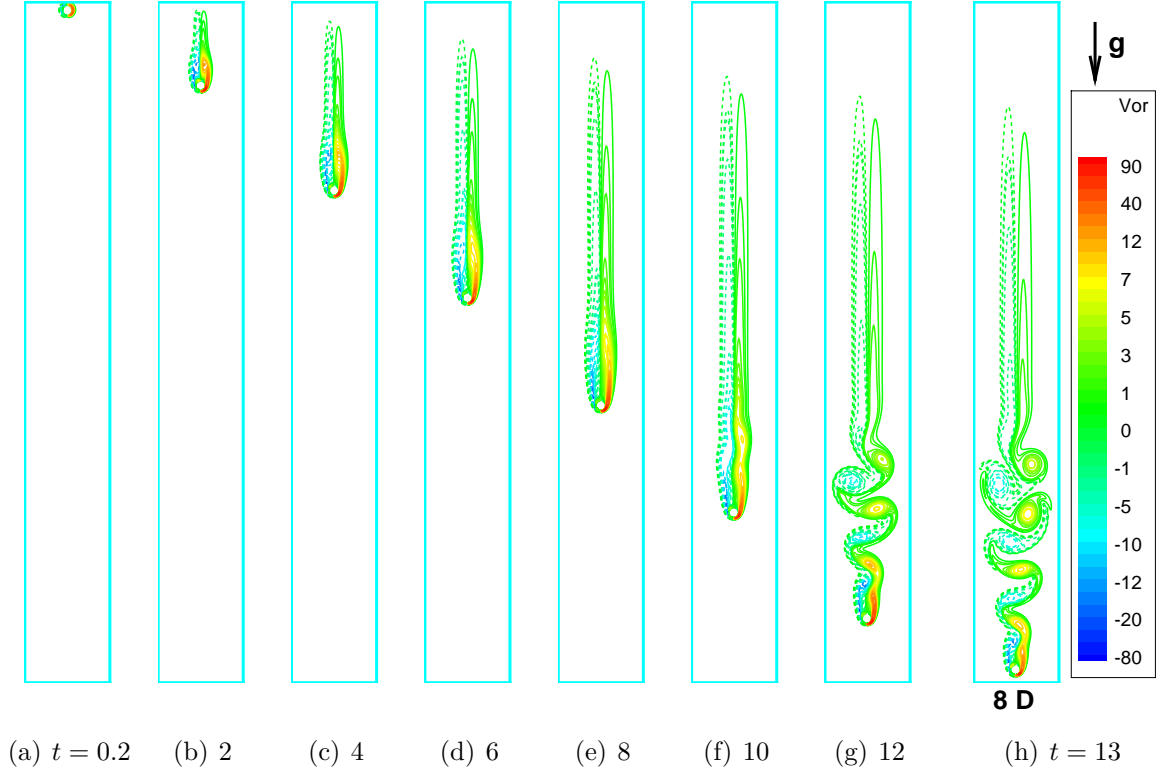


Figure 3.12: Vorticity isolines (dashed lines are used for negative values) of the falling cylinder in fully quiescent fluid, performed by the 2^{nd} -order solver, where free-slip boundary conditions are imposed at the surrounding walls, $g = -9.81 \text{ m/s}^2$, $\rho_b/\rho_f = 1.01$, $D = 0.005 \text{ m}$, $(x, y) \in [0, 0.04 \text{ m}] \times [0, 0.32 \text{ m}] = [0, 8D] \times [0, 64D]$, $\Delta t = 1.25 \times 10^{-4}$, the resolution is set to 512×4096 , the penalization parameter $\eta = 10^{-3}$, the filter parameter for denoising of the hydrodynamic coefficients is $\delta = 0.001$, $\nu = 8 \times 10^{-7} \text{ m}^2/\text{s}$ and $Re \approx 156$.

to 0.003 results in different falling regimes. Snapshots of vorticity isolines of the falling ellipse in different regimes are illustrated in Fig. 3.19. Other parameters which are used in the simulations are described as follows: The polar moment of inertia around the center of gravity is $J_{cg} = 0.25\pi ab(a^2 + b^2)\rho_b = 0.0157$, the initial position $(x_0, y_0) = (0.5L_x, L_y - 3a)$ and the initial angle of the major diameter with respect to the horizon is $\theta_0 = \pi/4$. The density ratio is set to $\rho_b = 1.538\rho_f$, the filter parameter for denoising of the hydrodynamic coefficients $\delta = 0.001$, the gravity in the y -direction $g = -9.81 \text{ m/s}^2$ and the penalization parameter is $\eta = 10^{-3}$. Isolines of the vorticity and the trajectory of the center of gravity corresponding to ellipse falling in the steady regime are illustrated in Fig. 3.15 at different instants. Isolines of the vorticity and the trajectory of the center of gravity corresponding to ellipse falling in the fluttering regime are illustrated in Fig. 3.16 at different instants from $t = 0.2$ up to $t = 25$. Isolines of the vorticity and the trajectory of the center of gravity corresponding to the falling ellipse in the tumbling regime are illustrated in Fig. 3.17 at different instants. A qualitative agreement of the (cg) trajectories in different falling

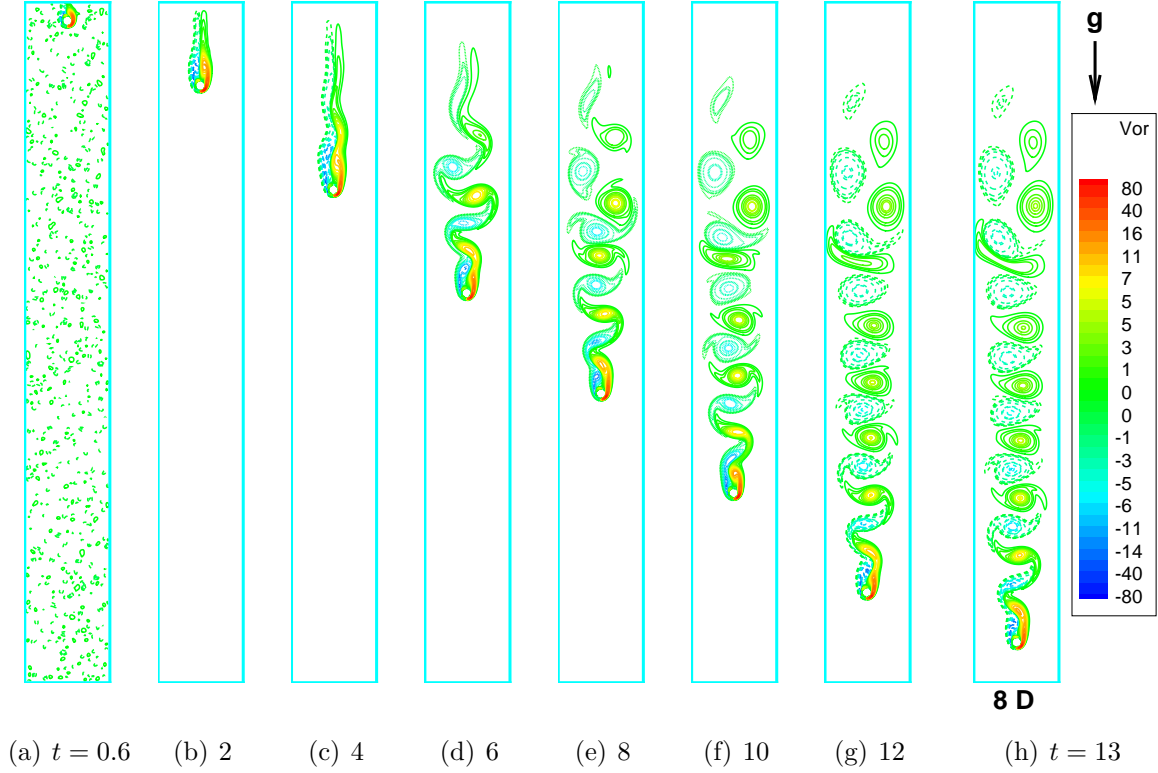


Figure 3.13: Vorticity isolines (dashed lines are used for negative values) of the falling cylinder in a slightly perturbed fluid, performed by the 4th-order solver, where free-slip boundary conditions are imposed at the surrounding walls, $g = -9.81 \text{ m/s}^2$, $\rho_b/\rho_f = 1.01$, $D = 0.005 \text{ m}$, $(x, y) \in [0, 0.04 \text{ m}] \times [0, 0.32 \text{ m}] = [0, 8 D] \times [0, 64 D]$, $\Delta t = 1.25 \times 10^{-4}$, resolution 4096×512 , penalization parameter $\eta = 5 \times 10^{-4}$, $\Delta t = 1.25 \times 10^{-4}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 8 \times 10^{-7} \text{ m}^2/\text{s}$ and $Re \approx 150$.

regimes with the simulations of Gazzola et al. [144] can be observed in Fig 3.18. The differences in the (cg) trajectories are due to the slightly different parameters we have used and the chaotic behavior of ellipse in the tumbling regime. The amplitude of the oscillations in the fluttering regime is also sensitive to the used parameters. The corresponding forces and velocity components of the falling ellipse in the fluttering regime are plotted in Fig. 3.20. A comparison of the first and second order filtering of the hydrodynamic forces is shown in Fig. 3.20 (a) - (b) and (c). As can be seen the second-order filtering is more efficient for denoising the hydrodynamic forces in comparison to the first-order filtering. The hydrodynamic coefficients in the fluttering regime show an oscillatory behavior with a principal frequency $f_1 \approx 0.24$. However, in the side force a harmonic frequency with $f_2 = 2f_1 \approx 0.48$ can be seen which is due to the shedding of the vortices. The chosen reference point in the simulation of the falling ellipse is the center of gravity (cg) for the calculation of the polar moment of inertia, rotation angle and the moment. This choice is advantageous for simplification of the Euler equation (2.29), by eliminating the torque due

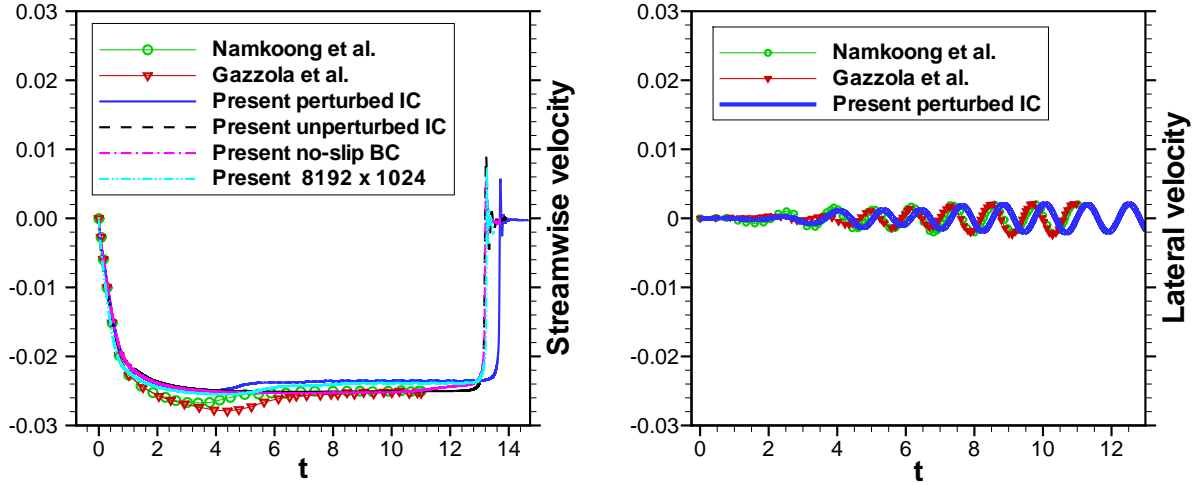


Figure 3.14: Comparison of the streamwise $u_{\text{streamwise}}$ and lateral u_{lateral} velocities of the falling cylinder via different methods/parameters with reference simulations. Symbols indicate the simulations performed by Gazzola et al. [144] (red triangles) and Namkoong et al. [119] (green circles). Solid and dashed lines represent the results with the proposed algorithm on 4096×512 grid point with penalization parameter $\eta = 10^{-3}$, respectively performed by, the 4th-order solver with a perturbed IC and free-slip BC (blue solid), the 2nd-order solver with unperturbed IC and free-slip BC (black dashed), the 2nd-order solver with unperturbed IC and no-slip BC (purple dash-dot) and the 2nd-order solver with perturbed IC and free-slip BC (cyan dash-dot-dot) on the finest resolution 8192×1024 with penalization parameter $\eta = 10^{-4}$.

to buoyancy. In Chapter 4 numerical simulation of swimming fishes will be considered. For the simulations of the swimming fish ($\rho_b = \rho_f$) the buoyancy is equal to zero. Thus without the need for evaluation of the torque due to the body forces in Euler equation (2.29), the reference point can move to the head, which is more suitable for the construction of the fish geometry and its kinematics. The geometry and kinematics of the fish are calculated by Eqs (4.10), (4.12) and (4.14), where starting by the information of the head as initial conditions is advantageous.

3.5 Conclusion

In this chapter the ability of the proposed algorithm for simulation of solid bodies interacting with two-dimensional incompressible flows is examined. For introducing a solid body in fluid flow, the volume penalization method is applied to the Navier–Stokes equations as a forcing term. Even if the penalization method is shown to have between first and second order accuracy in space, an advantage of this method is that the evaluation of the hydrodynamic coefficients is straightforward. Proper denoising of the hydrodynamic coefficients is crucial in dealing with fluid–solid interaction problems via the volume penalization

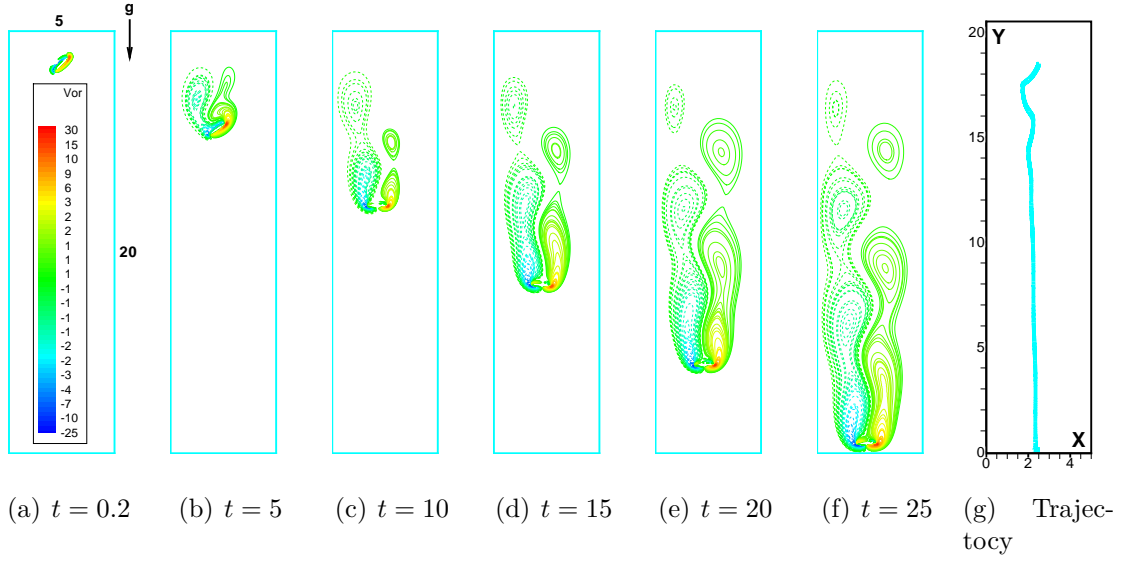


Figure 3.15: Vorticity isolines (dashed lines are used for negative values) of the falling ellipse in the steady regime, where resolution of the grid is $Im \times Jm = 513 \times 2049$, $(x, y) \in [0, 5L] \times [0, 20L]$, $L = 2a = 1$, $J^* = 0.16$, $\rho_s/\rho_f = 1.538/1.0$, $g = -9.81$, $a/b = 0.5/0.1$, $X_0^{cg} = L_x/2$, $Y_0^{cg} = L_y - 3a$, $\theta_0 = \pi/4$, $\delta_{\text{filter}} = 10^{-3}$, $\eta = 10^{-3}$, $\nu = 0.03$ and $Re \approx 15$.

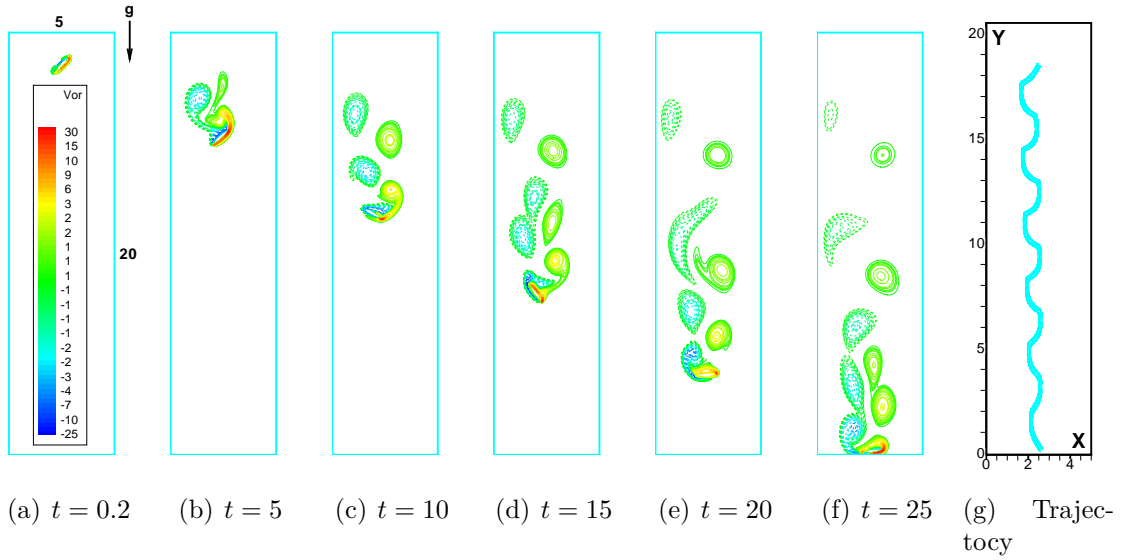


Figure 3.16: Vorticity isolines (dashed lines are used for negative values) of the falling ellipse in the fluttering regime, where resolution of the grid is $Im \times Jm = 513 \times 2049$, $(x, y) \in [0, 5L] \times [0, 20L]$, $L = 2a = 1$, $J^* = 0.16$, $\rho_s/\rho_f = 1.538/1.0$, $g = -9.81$, $a/b = 0.5/0.1$, $X_0^{cg} = L_x/2$, $Y_0^{cg} = L_y - 3a$, $\theta_0 = \pi/4$, $\delta_{\text{filter}} = 10^{-3}$, $\eta = 10^{-3}$, $\nu = 0.01$ and $Re \approx 46$.

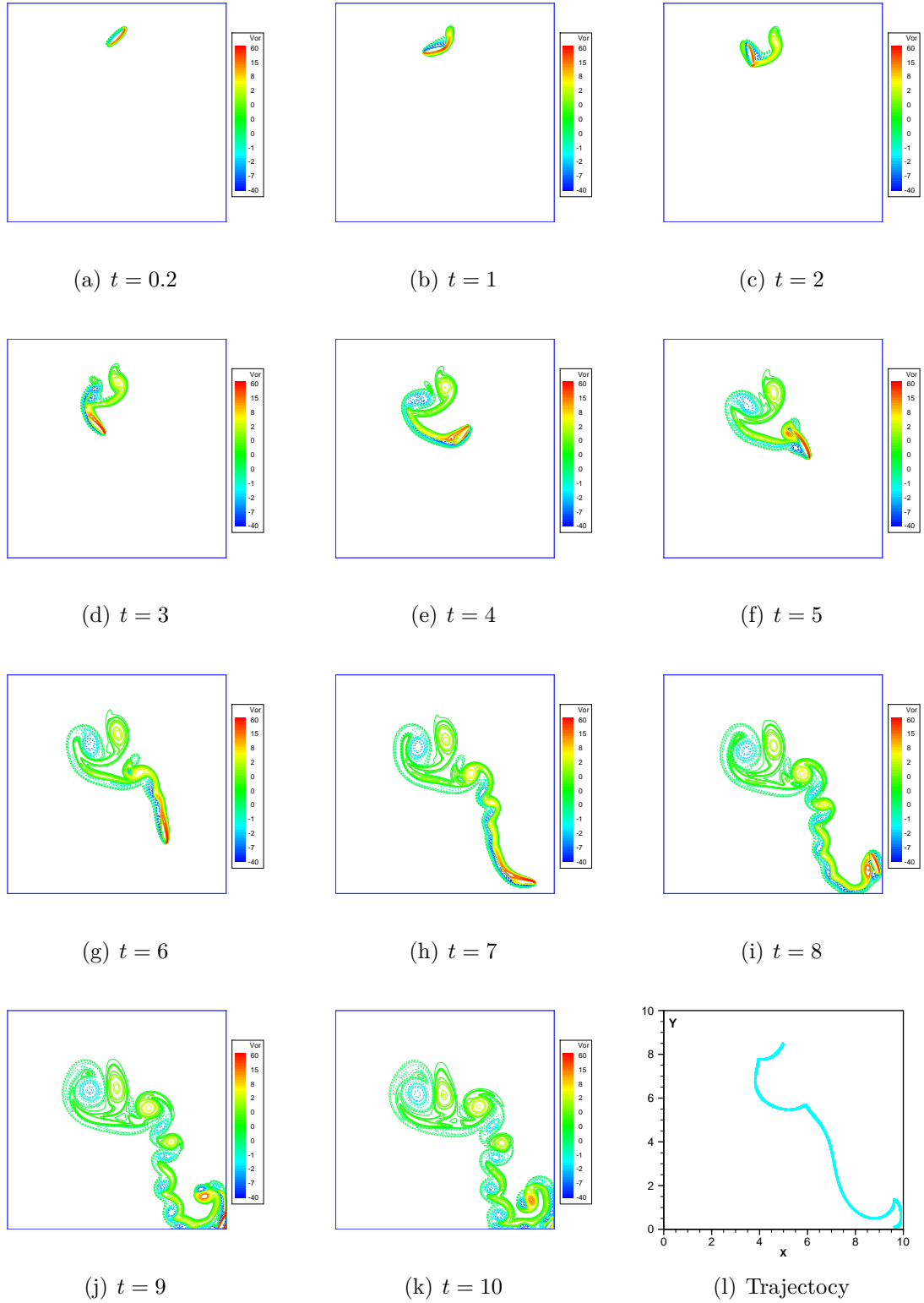
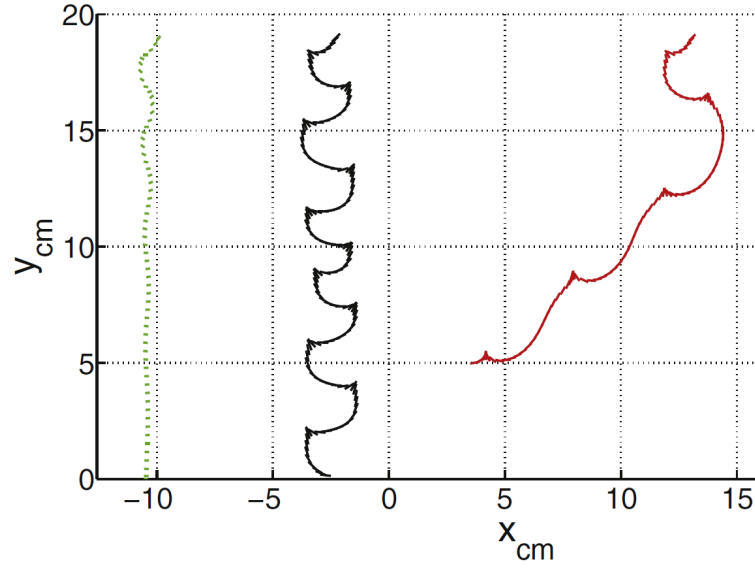


Figure 3.17: Vorticity isolines (dashed lines are used for negative values) of the falling ellipse in the tumbling regime, where resolution of the grid is $Im \times Jm = 2049^2$, $(x, y) \in [0, 10] \times [0, 10]$, $J^* = 0.16$, $\rho_s/\rho_f = 1.538/1.0$, $g = -9.81$, $a/b = 0.5/0.1$, $X_0^{cg} = L_x/2$, $Y_0^{cg} = L_y - 3a$, $\theta_0 = \pi/4$, $\delta_{\text{filter}} = 10^{-3}$, $\eta = 10^{-3}$, $\nu = 0.003$ and $Re \approx 153$.



(a) Picture from Gazzola et al. [144]. Coordinates are reported in cord lengths: (green) steady falling regime ($J^* = 0.146$, $Re = 100$, $H/L = 1/4$ and $\rho_s/\rho_f = 1.1$), (black) fluttering regime ($J^* = 0.16$, $H/L = 1/5$, $\rho_s/\rho_f = 1.538$, $\nu = 6.33 \times 10^{-3}$, $u_t = 1.45$ and $Re = 1147$,) and (red) tumbling regime ($J^* = 0.146$, $Re = 1000$, $H/L = 1/4$ and $\rho_s/\rho_f = 1.1$).

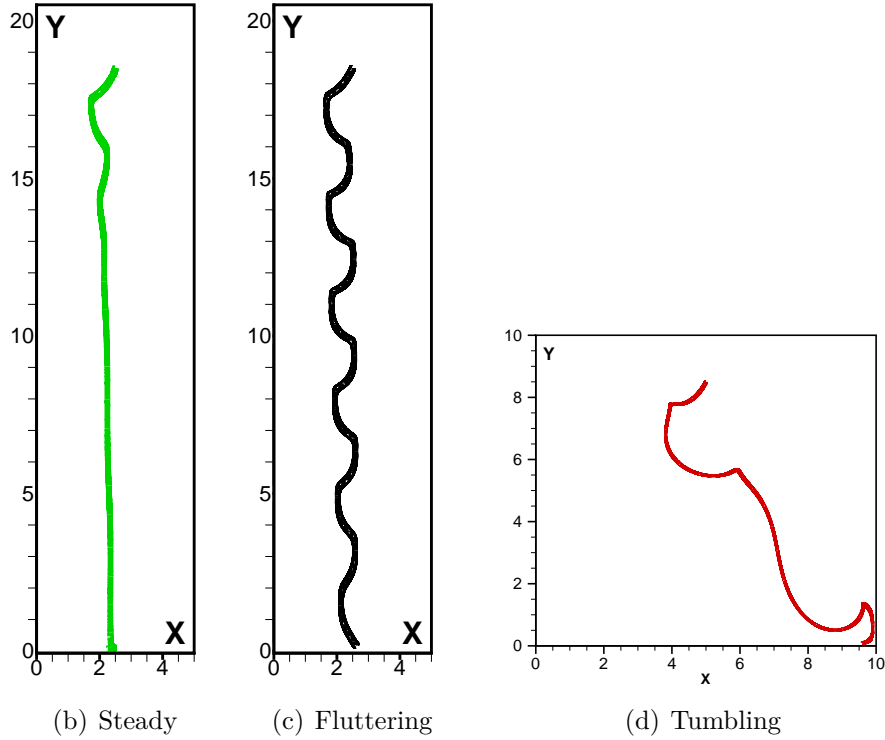
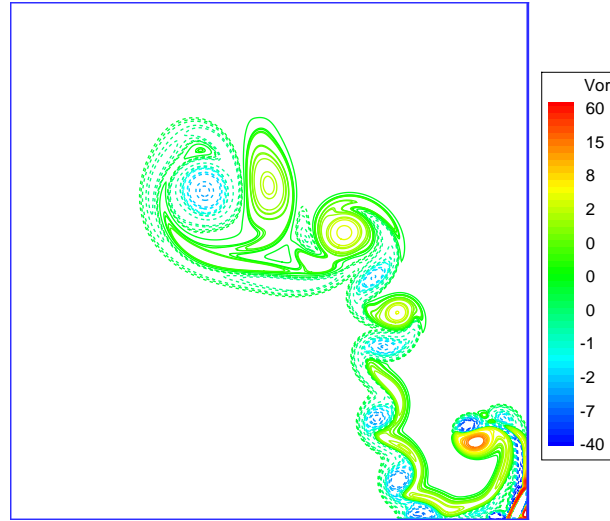
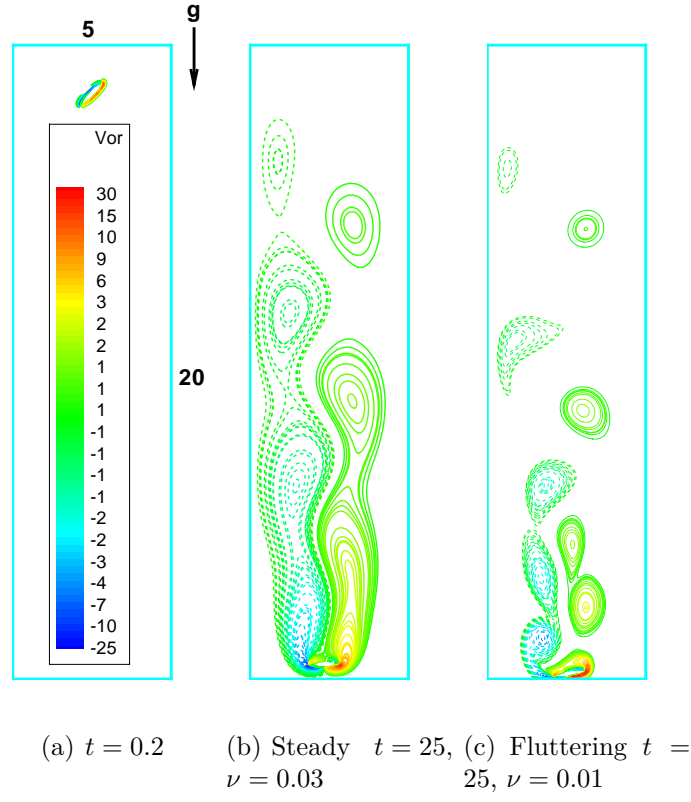


Figure 3.18: Comparisons of (cg) trajectories of the falling ellipse, obtained in the present investigation, with those of Gazzola et al. [144].

method. Validation of the developed method shows the efficiency and expected accuracy of the algorithm for a variety of fluid–solid interaction problems. Some perspectives for future works are adding a multiresolution analysis to the algorithm for grid adaptation, parallelization and extension to three dimensions.



(d) Tumbling $t = 9$, $\nu = 0.003$

Figure 3.19: Vorticity isolines (dashed lines are used for negative values) of the falling ellipse in different regimes, where $J^* = 0.16$, $\rho_b/\rho_f = 1.538$, $g = -9.81$, $a/b = 0.5/0.1$, $X_0^{cg} = L_x/2$, $Y_0^{cg} = L_y - 3a$, $\theta_0 = \pi/4$, $\delta_{\text{filter}} = 10^{-3}$ and $\eta = 10^{-3}$.

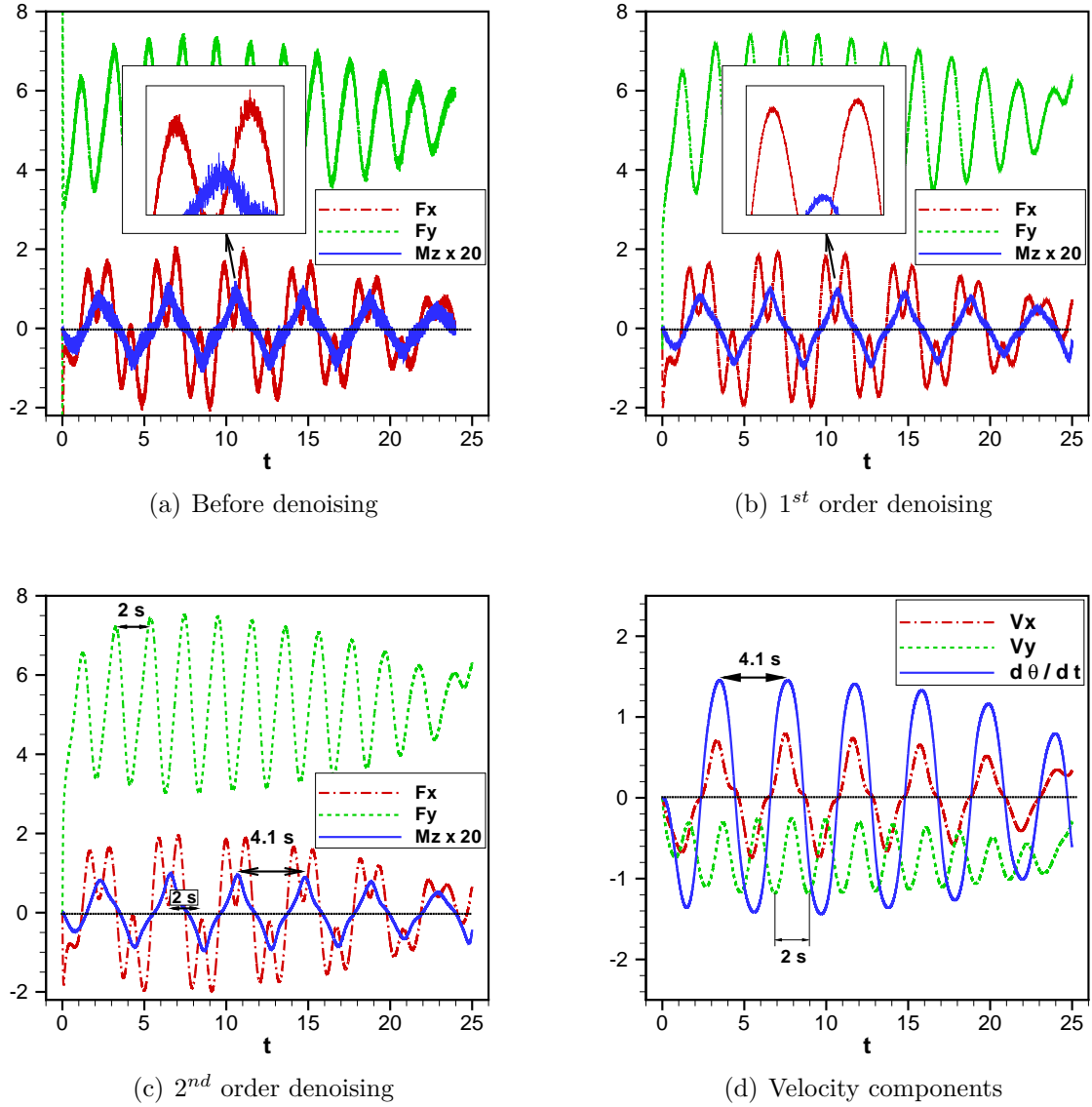


Figure 3.20: (a) Hydrodynamic coefficients of a falling ellipse in the fluttering regime, where $J^* = 0.16$, $\rho_b/\rho_f = 1.538$, $a/b = 1/5$ and $\nu = 0.01$ before denoising. (b) After applying the first-order filter (2.25) with $b = 0$ and $\alpha = 0.2$. (c) After applying the second-order filter via Eqs. (2.25) and (2.26) with $\delta = 0.001$. (d) The corresponding velocity components.

Chapter 4

Two-dimensional simulations of fish-like swimming

“In spite of a common fascination, even obsession, with features, we too often forget to appreciate them in their natural setting, gracing the wild creatures around us.”

Thor Hanson (2011) features:

The evolution of a natural miracle

First we recall a brief introduction to fishes and locomotion types from Wikipedia (the free encyclopedia) [165]-[166]. A fish is any gill-bearing aquatic animal that lack limbs with digits. They can be divided into *bony fish (osteichthyes)*, *cartilaginous* and *hagfish (lampreys)*. Fishes exhibit greater species diversity than any other group of vertebrates with about 32,000 species [165]. Some examples are shown in Fig. 4.1. Their length ranging from 1 cm to 18 m. Similarly to the aerodynamics of flight, swimming requires to overcome the drag by producing thrust by the swimmer. Unlike flying, however, swimming animals do not necessarily need to actively exert high vertical forces because the effect of buoyancy can counter the downward pull of gravity, allowing these animals to float without much effort. Fish swims by exerting force against the surrounding water. This is normally done by the fish contracting muscles on either side of its body in order to generate moving waves from head to tail, generally getting larger as they go toward the tail [166]. The resultant force exerted on the water by such motion generates a backward force (even oscillatory) which in turn pushes the fish forward. In straight swimming the time average of the resultant lateral force is zero. Most fishes generate thrust by using lateral movements of their body and fins. But some fishes swim mainly using their median and paired fins. The latter group gain manoeuvrability but they cannot swim as fast as fishes using their bodies and

caudal fins. In general these movements can be divided into *undulatory* and *oscillatory* motions. Following Breder [6] mechanisms of swimming (locomotion), using body-caudal fins, are divided to five groups that differ in the fraction of the body that is displaced laterally.

Anguilliform: This mechanism can be observed in some long, slender fish-eels, where there is little increase in the amplitude of the flexion wave as it passes along the body.

Sub-carangiform: In this case there is a more marked increase in wave amplitude along the body with the vast majority of the work being done by the rear half of the fish. In general, the fish body is stiffer, leading to higher speed but reduced maneuverability. The Trout which is demonstrated in Fig. 4.1 (b) use sub-carangiform of locomotion.

Carangiform: Fishes of this group are stiffer and faster-moving than the previous groups. The vast majority of movement is concentrated in the very rear of the body and tail. Carangiform swimmers generally have rapidly oscillating tails.

Thunniform: The next-to-last group is reserved for the high-speed long-distance swimmers, like tuna. Hawkins et al. [94] show that the thunniform locomotion is an autapomorphy of the tunas. Here, virtually all the lateral movement is in the tail and the region connecting the main body to the tail (the peduncle). The tail itself tends to be large and crescent shaped.

Ostraciiform: These fishes have no appreciable body wave when they employ caudal locomotion. Only the tail fin itself oscillates (often very rapidly) to create thrust. This group includes Ostraciidae.

Median-paired fin propulsion: Not all fishes fit comfortably in the five above groups. Ocean sunfish, for example, have a completely different locomotion system, or many small fishes use their pectoral fins for swimming as well as for steering and dynamic lift. Fishes with electric organs, such as those in Gymnotiformes, swim by undulating their fins while keeping the body still, presumably so to not disturb the electric field that they generate [166]. Some locomotion models and body types are illustrated in Fig. 4.2. Three main parts of the body of the fishes are head, trunk and tail, an example with the external organs is illustrated in Fig. 4.3. Different types of fins like dorsal, ventral, anal, pectoral and caudal are shown in Fig. 4.4 for a haddock.

4.1 Physical definitions

In this section we introduce some dimensionless parameters, frequently used in the literature to quantify the swimming of fish-like animals due to undulatory movement. The Reynolds number is defined as

$$Re = \frac{\bar{U}L}{\nu}$$

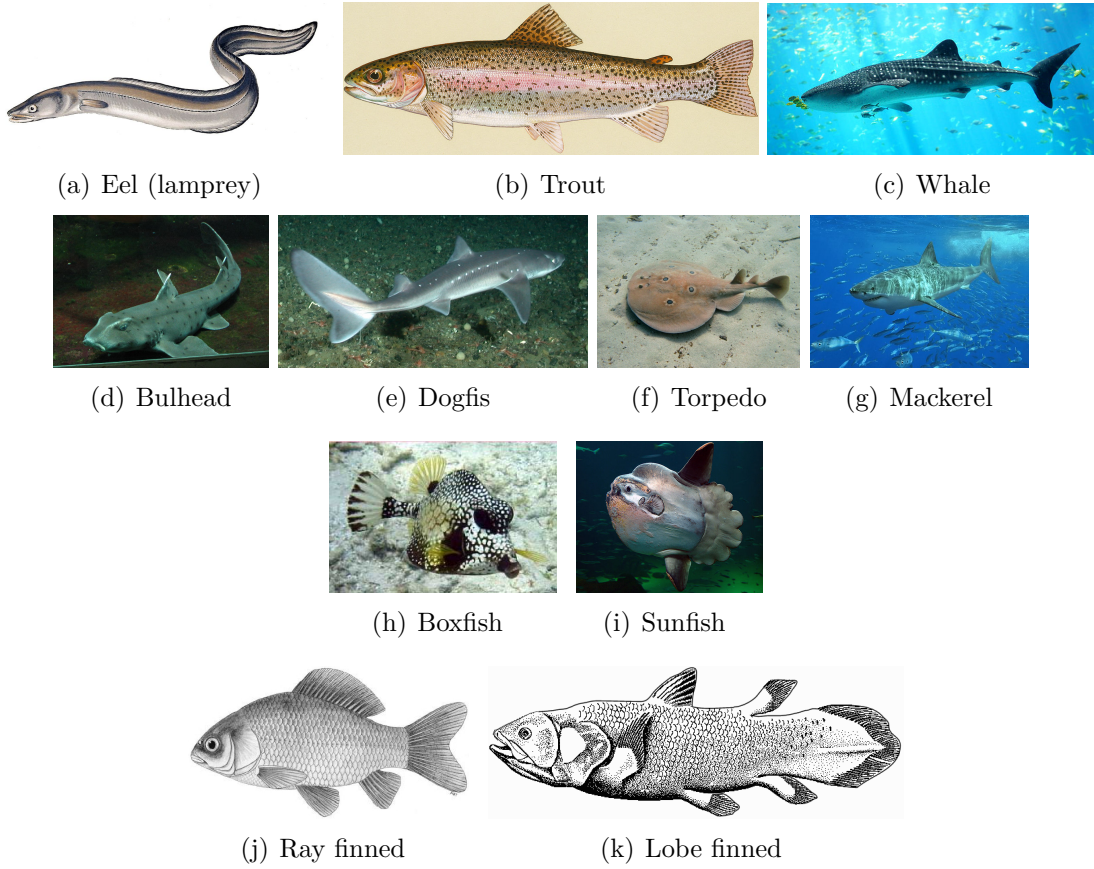


Figure 4.1: Different types of fishes. Cartilaginous fishes (c-g), pictures are taken from [165].

where \bar{U} is the average swimming speed, L is the length of the swimming fish, see Fig. 4.5 (a), and ν represents the kinematic viscosity. Lord Rayleigh [4] was the first to use the Strouhal number, previously defined by Strouhal [2], to quantify in a proper dimensionless fashion the frequency of vortex shedding behind a bluff body. A decade later, this definition was eventually changed by Bénard [7] to be the inverse of Rayleigh's suggestion:

$$St = f \frac{d}{U}$$

where f is the frequency of vortex shedding, d is the diameter of the bluff body and U is the free-stream velocity. The Strouhal number is intimately linked to the arrangements of vortices in the wake as already pointed out by Rayleigh. Von Kármán [3] showed that two infinite rows of point vortices are always unstable unless their spacing ratio has a particular value $b/a = 0.281$ (see Fig. 4.6 (a)). In the context of swimming, the Strouhal number has been introduced within two innovative papers by Triantafyllou et al. [51, 59]. Following

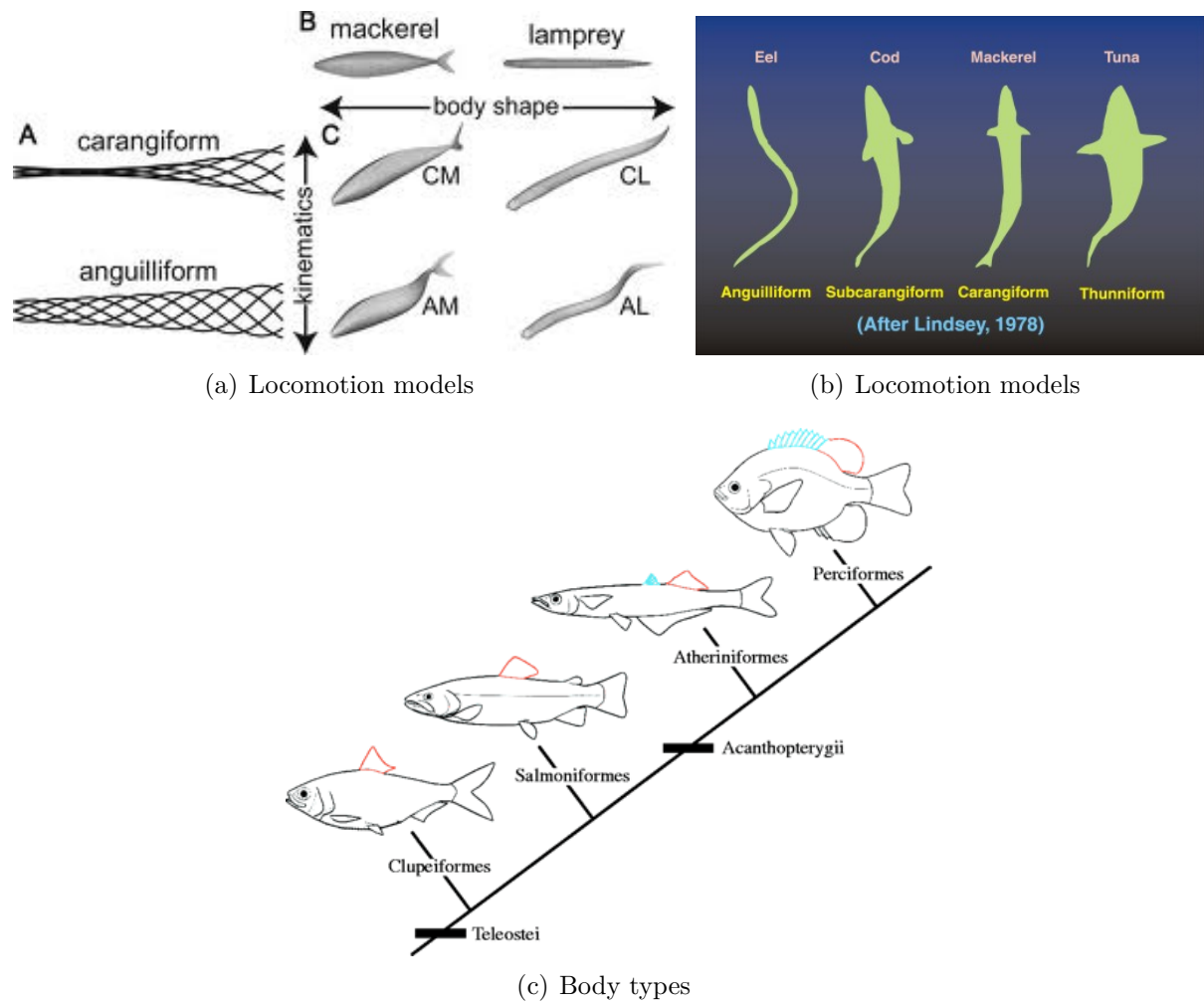


Figure 4.2: Locomotion models and body types

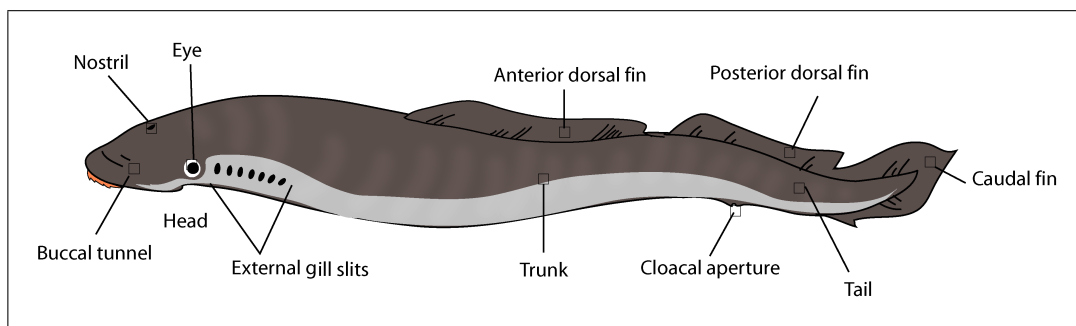


Figure 4.3: Basic external anatomy of a lamprey, picture is taken from [165].

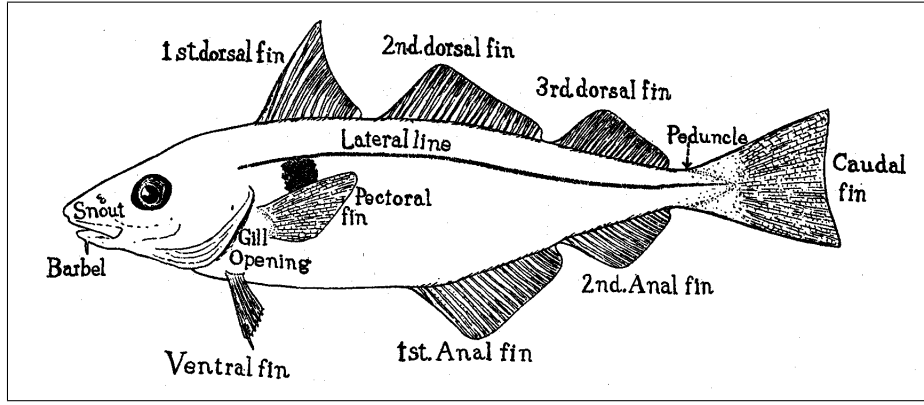


Figure 4.4: The haddock, a type of cod, is ray-finned. Pectoral fins (paired), ventral fins (paired), dorsal fin (three), adipose fin, anal fin (two) and caudal (tail) fin (one) are illustrated, picture is taken from [165].

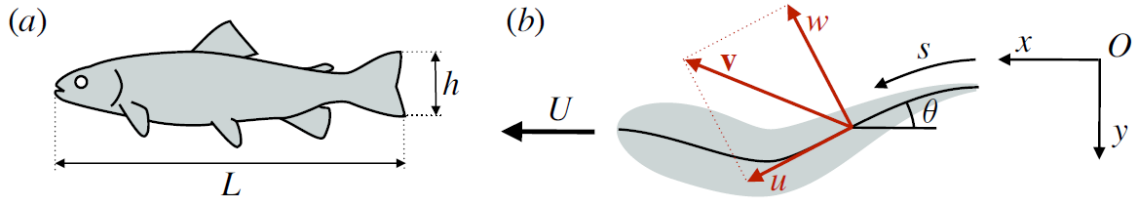


Figure 4.5: Dimensions considered for a typical swimming fish: (a) side view and (b) top view, picture is taken from Eloy [157].

Eloy [157] the Strouhal number is defined as:

$$St = f \frac{A}{\bar{U}}$$

where f is the tail-beat frequency, A is the peak-to-peak amplitude at the tail tip and \bar{U} is the average swimming speed. The argument of Triantafyllou et al. [59, 51] relies on the observation that the wake behind a swimming fish resembles the Bénard-von Kármán (BvK) vortex street observed behind bluff bodies except that the sign of vortices are inverted, resulting in a reverse Bénard-von Kármán (rBvK) street (see Fig. 4.6 (b)). In the BvK street, the average flow exhibits a deficit of velocity compared to the free stream U , indicating that longitudinal momentum has been lost and that a drag force is exerted on the bluff body (see Fig. 4.6 (c)). However, swimming animals are self-propelled and therefore no net drag nor thrust is exerted on average when they swim at constant speed. The resulting rBvK wake is therefore momentumless and exhibits on average a jet around the centerline surrounded by a region of counterflow (see Fig. 4.6 (d)). In Fig. 4.7 a schematic three-dimensional views of the BvK (a) and rBvK vortex streets (b), corresponding to the two-dimensional views of Fig. 4.6 is illustrated. In the case of steady swimming, the thrust

has to compensate the drag D on average such that

$$\langle T \rangle = D$$

where $\langle \cdot \rangle$ angle brackets denote time average. Hence, the role of viscosity being limited to setting the drag, the only relevant parameters are the added mass at the tail tip

$$m_a = \frac{\pi}{4} \rho h^2,$$

the average swimming velocity \bar{U} and the drag D . Out of these three parameters, a unique dimensionless quantity can be constructed which measures the ratio between the drag D and the typical thrust $m_a \bar{U}^2$. The resulted dimensionless number is called the Lighthill [28] number

$$Li = \frac{\pi D}{2m_a \bar{U}^2} = \frac{S}{h^2} C_d$$

where S is the total surface of the fish (or wetted surface) and C_d is the drag coefficient such that

$$C_d = \frac{D}{0.5 \rho \bar{U}^2 S}$$

In the same manner the thrust coefficient is defined as $C_t = T/(0.5 \rho \bar{U}^2 S)$. The only relevant parameter to the swimming problem is Lighthill number Li which gathers informations on the geometry of the swimming animal (through the shape ratio S/h^2) and on the Reynolds number (through the drag coefficient C_d). The optimal motion of the tail will thus be a function of Lighthill number Li alone. This is in contrast to bluff body wakes where the Strouhal number is a function of Re , as it has been shown by Rayleigh. Another useful parameter in quantifying the hydrodynamic efficiency of the swimming, is the slip ratio commonly defined as

$$S_r = \frac{U}{V_p} = U \frac{k}{\omega}$$

where k denotes the wavenumber, ω is the angular frequency of the oscillations and $V_p = \omega/k$ is the velocity of the passing wave (phase speed) due to undulatory movement of the body which is always greater than the swimming speed \bar{U} , i.e., ($SR < 1$).

The swimming number is defined by Gazzola et al. [162] as follows:

$$Sw = f \frac{AL}{\nu}$$

which is resulted from multiplication of Reynolds number by Strouhal number, i.e.,

$$Sw = Re St$$

The most interesting point in this nondimensional number is the elimination of swimming speed \bar{U} which is present in all previously defined numbers. In the context of swimming, velocity is an output of the imposed kinematics on the body, which is unknown by default before starting the simulation. In some sense, swimming number is reminiscent of the Péclet number $Pe = Re Pr$ defined in heat transfer, which is resulted from multiplication of Reynolds number by Prandtl number. Then a scaling law presented by Gazzola et al. [162] for quantification of swimming as follows:

$$Sw \sim \begin{cases} Re^{3/4} & Re \leq 2500 \\ Re & Re > 2500 \end{cases} \quad (4.1)$$

where $Re_c \in [2000, 7000]$ represents the critical Reynolds that transition from laminar to turbulent regime takes place. For turbulent swimming the Strouhal number is approximately constant $St \simeq 0.3$ but for laminar swimming they propose $St \sim Re^{-1/4}$. For more details we refer to Gazzola et al. [162].

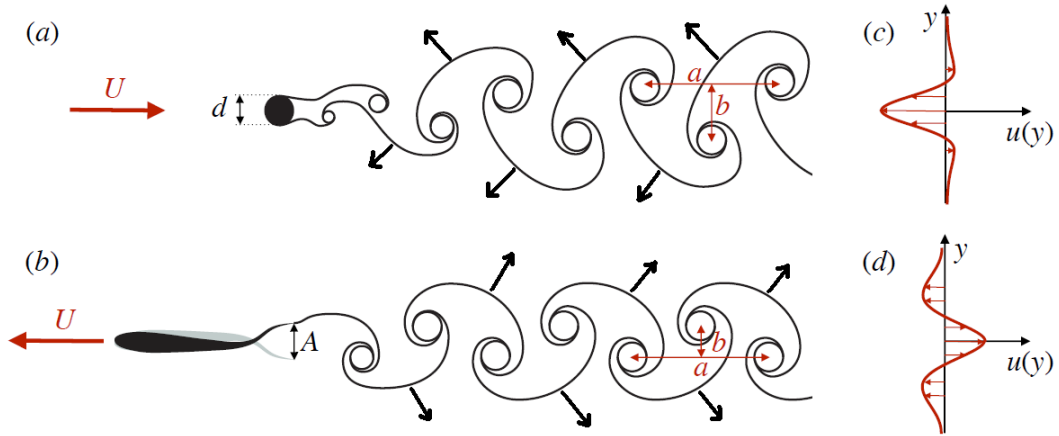


Figure 4.6: (a) Schematic view of the Bénard-von Kármán (BvK) vortex street behind a circular cylinder. (b) The reverse BvK (rBvK) vortex street in the backside of a swimming fish. (c) The average velocity difference $u(y)$ from the mean flow U in the far wake is a jet toward the cylinder. (d) In the case of swimming $u(y)$ is backward oriented in the center line. Both of these jets are surrounded by a region of counterflow. In a stable configuration of vortices each dipole creates a small jet represented by black vectors, pictures are taken from Eloy [157] with a slight modification.

4.1.1 Efficiency measurement

An important issue in studying the fish swimming, is to classify the hydrodynamic efficiency of the movement. There are different definitions depending to the case and the purpose of the study. We will cite some of them in the following. We denote by $\sigma'_{ij} = \mu(u_{i,j} + u_{j,i})$ the viscous stress tensor. Following Bergmann and Iollo [145] the power required for the

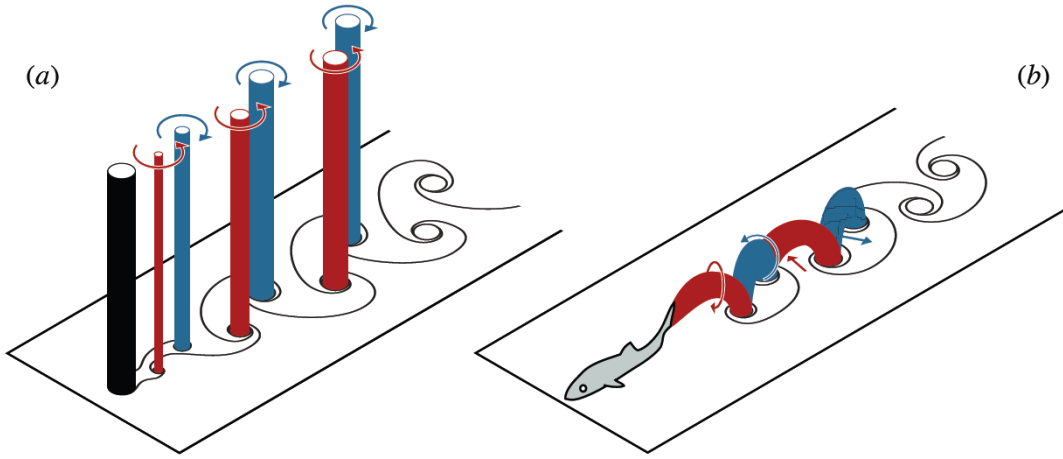


Figure 4.7: Schematic three-dimensional views of the (a) BvK and (b) rBvK vortex streets, corresponding to the two-dimensional views of Fig. 4.6, picture is taken from Eloy [157].

swimming is defined as:

$$P(t) = - \int_{\partial\Omega_s} p \mathbf{u} \cdot \mathbf{n} dS + \int_{\partial\Omega_s} (\sigma'_{ij} \cdot \mathbf{n}) \cdot \mathbf{u} dS \quad (4.2)$$

Since in the present investigation the mesh is not body fitted, $P(t)$ cannot be computed by Eq. (4.2) in a straightforward manner. By integrating the scalar product of the momentum equations and the velocity vector over the fluid domain Ω_f , following Bergmann and Iollo [145] the total instantaneous power delivered to the fluid can be written as:

$$P(t) = \frac{\partial}{\partial t} \int_{\Omega_f} \rho_f \frac{|\mathbf{u}|^2}{2} d\Omega + \mu \int_{\Omega_f} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) : \frac{\partial u_i}{\partial x_j} d\Omega \quad (4.3)$$

where $|\mathbf{u}|^2 = \mathbf{u} \cdot \mathbf{u} = u^2 + v^2 + w^2$ and Ω_f denotes the spatial region occupied by the fluid. The power required to swim is then equal to the rate of change (or temporal variation) of the kinetic energy in the flow domain plus the power dissipated by viscosity. The required energy for a fish to travel a given distance between t_1 and t_2 is

$$E = \int_{t_1}^{t_2} P(t) dt \quad (4.4)$$

By denoting the mean power required for a considered steady periodic swimming at a velocity U by P_{sps} and the mean power needed to tow the same rigid body at the same velocity U by P_{tow} . Following Barrett et al. [71], the propulsive index I_p is defined as,

$$I_p = \frac{P_{tow}}{P_{sps}}$$

Both of the powers, i.e., P_{sps} and P_{tow} , are computed from Eq. (4.3). Drag reduction can be achieved if $I_p > 1$. Following Gazzola et al. [155] efficiency is defined as:

$$\eta = \frac{E_{\text{useful}}}{E_{\text{flow}}}$$

where E_{useful} is the kinetic energy of the fish:

$$E_{\text{useful}} = \frac{1}{2}m\bar{U}^2$$

in which \bar{U} is the mean forward velocity of the fish during the swimming and m is the mass of the fish. The term E_{flow} represents the total energy delivered to the fluid during the swimming and can be computed by time integration of Eq. (4.3). Following Eloy [157] another parameter to estimate the swimming performance, is the Froude efficiency:

$$\eta_{fr} = \frac{\langle T\bar{U} \rangle}{D\bar{U} + \langle E \rangle} \quad (4.5)$$

which expresses the ratio between the average useful power $\langle T\bar{U} \rangle = D\bar{U}$ and the total power spent for swimming.

4.2 Modeling of the swimmer shape

A symmetric shape is the first choice to start the parameterization of the swimmer body. A class of swimmers shape can be described by a hydrofoil. One method to parameterize a hydrofoil shape is the Kutta–Joukowski transform. In this transformation, a circle with radius $r_c = 1$ in original plane (space), defined by the complex number $\zeta = \eta + i\theta$, change into an airfoil profile, defined by the complex number $z = x + iy$, in the transformed plane (see Fig. 4.8). The transform is defined as follows:

$$z = \zeta + \frac{\lambda^2}{\zeta}, \quad \lambda \in \mathbb{C}$$

The circle must enclose the point $\zeta = -1$ (where the derivative is zero) and intersects the point $\zeta = 1$. This can be achieved for any allowable center position $(\eta_c + i\theta_c)$ by varying the radius of the circle. Since this hydrofoil presents a cusped trailing edge, following Bergmann and Iollo [145] the Kármán-Trefftz transform can be applied to create more realistic shapes. Even with the use of the Joukowski transform the cusp in the trailing edge can be eliminated by slightly thickening the trailing edge via two methods; either by directly modifying the coordinates of the points at the body surface, or in the process of determining the mask function χ , filtering of the mask function $\bar{\chi}$ can smooth the trailing edge.

The 4-digit NACA-00xx airfoils can also be used to produce a symmetric hydrofoil. The formula for the thickness xx of the foil, with $T=xx/100$, is

$$y_t(x) = 5T(0.2969x^{0.5} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4)$$

where the cord length is nondimensionalized to be in $x \in [0, 1]$.

Following the works of Carling et al. [67], Kern and Koumoutsakos [105] and Gazzola et al. [144] the geometry of a two-dimensional swimmer can be characterized by the half width $w(s)$ of the body along its (midline) arclength (s). In their study the half width $w(s)$ is defined as:

$$w(s) = \begin{cases} \sqrt{2w_h s - s^2} & 0 \leq s < s_b \\ w_h - (w_h - w_t)\left(\frac{s-s_b}{s_t-s_b}\right)^2 & s_b \leq s < s_t \\ w_t \frac{L-s}{L-s_t} & s_t \leq s \leq L \end{cases} \quad (4.6)$$

where L is the body length, $w_h = s_b = 0.04L$, $s_t = 0.95L$ and $w_t = 0.01L$. In Kern and Koumoutsakos [105], the thickness reduction from head to tail is linear instead of quadratic for the two dimensional cases. Here we implemented the same modification like as Gazzola et al. [144]. In the present study the Joukowski transform is used in preliminary simulations because of its simplicity and efficiency and proper accuracy. Kármán-Trefftz transform presented in [145] can also be used. In later simulations of the present study, Eq. (4.6) is used to determine the shape of the swimming fish.

4.2.1 Backbone deformation law for straight swimming

To define a periodic swimming law, the idea is to deform the backbone of the straight fish (defined by $0 \leq x \leq l$ and $y = 0$), see Fig. 4.8 (c), to fit a given curve $y(x, t)$ while keeping the backbone length fixed. Let s be the arclength over the curvilinear coordinate of the deformed backbone ($0 \leq s \leq l$). By choosing $s = 0$ at the head $\mathbf{x} = \mathbf{x}_0$, following Bergmann and Iollo [145], for a given arclength s the abscissa $x(s)$ is found by inverting the arclength integral:

$$s(x) = \int_{x_0}^x \sqrt{1 + \left(\frac{\partial y(x', t)}{\partial x'}\right)^2} dx' \quad (4.7)$$

Therefore in discrete space, for points uniformly distributed on the backbone curve, we have:

$$\Delta x = \frac{\Delta s}{\sqrt{1 + \left(\frac{\partial y}{\partial x}\right)^2}} \quad (4.8)$$

where $\Delta s = l/(n-1)$. The corresponding $y(x, t)$ coordinate can then be computed according to Eq. (4.9). As described by Barrett et al. [71], one of the frequently used modes of

propulsion, imposed over the backbone of the fish in a forward gait is

$$y(x(s), t) = a(x) \sin \left(2\pi(x/\lambda \pm ft) \right) \quad (4.9)$$

where $a(x)$ is the backbone envelope (see Fig. 4.9, Fig. 4.10 and Fig. 4.12). For a periodic undulatory swimming the envelope is given by:

$$a(x) = a_0 + a_1x + a_2x^2 \quad , \quad x \in [0, l_{\text{fish}}]$$

While the length of the fish is always equal to l , the Cartesian abscissa is $x(l) < l$, except for the straight configuration where $x(l) = l$. The motion is defined by a constant phase speed

$$V_p = \lambda f = \frac{\omega}{k} = \frac{U}{S_r}$$

where λ and f denote respectively the wavelength and the frequency of the oscillations. The wavelength λ is not necessarily equal to the body length as in Deng et al. [110]. The wavy motion affects the swimmer from the head to the tail and can be centered at the head of the fish, as in Deng et al. [110] or at a predefined percentage of the body length like in Zhu et al. [83]. The parameters a_0 , a_1 and a_2 can thus be adjusted in order to impose a maximal tail amplitude $A/2 = a_0 + a_1 + a_2$, which is an important parameter for the locomotion efficiency according to Lighthill [26]. Note that in practice for starting a wavy motion a progressive increase takes place, during a period T from the initial straight shape ($y(x, t) = 0$) to the final amplitude given by Eq. (4.9) (see Fig. 4.17). By choosing $l = 1$ over the backbone of the fish and fixing the position of the center of gravity as a function of the shape, the tail and the head of the swimmer move over an 8 shape or draw a wing of a butterfly, see Fig. 4.11 and Fig. 4.12. Another important parameter to quantify the kinematics of the swimming fish is the incident angle at the tail (i.e., the angle between the tail and the swimming direction), which is given in the vicinity of the tail tip by a harmonic function

$$\theta(t) = \arctan \left(\frac{dy}{dx} \right) = \theta_0 \cos(\omega t)$$

where θ_0 is the maximum incident angle at the tail, see Fig. 4.5 (b).

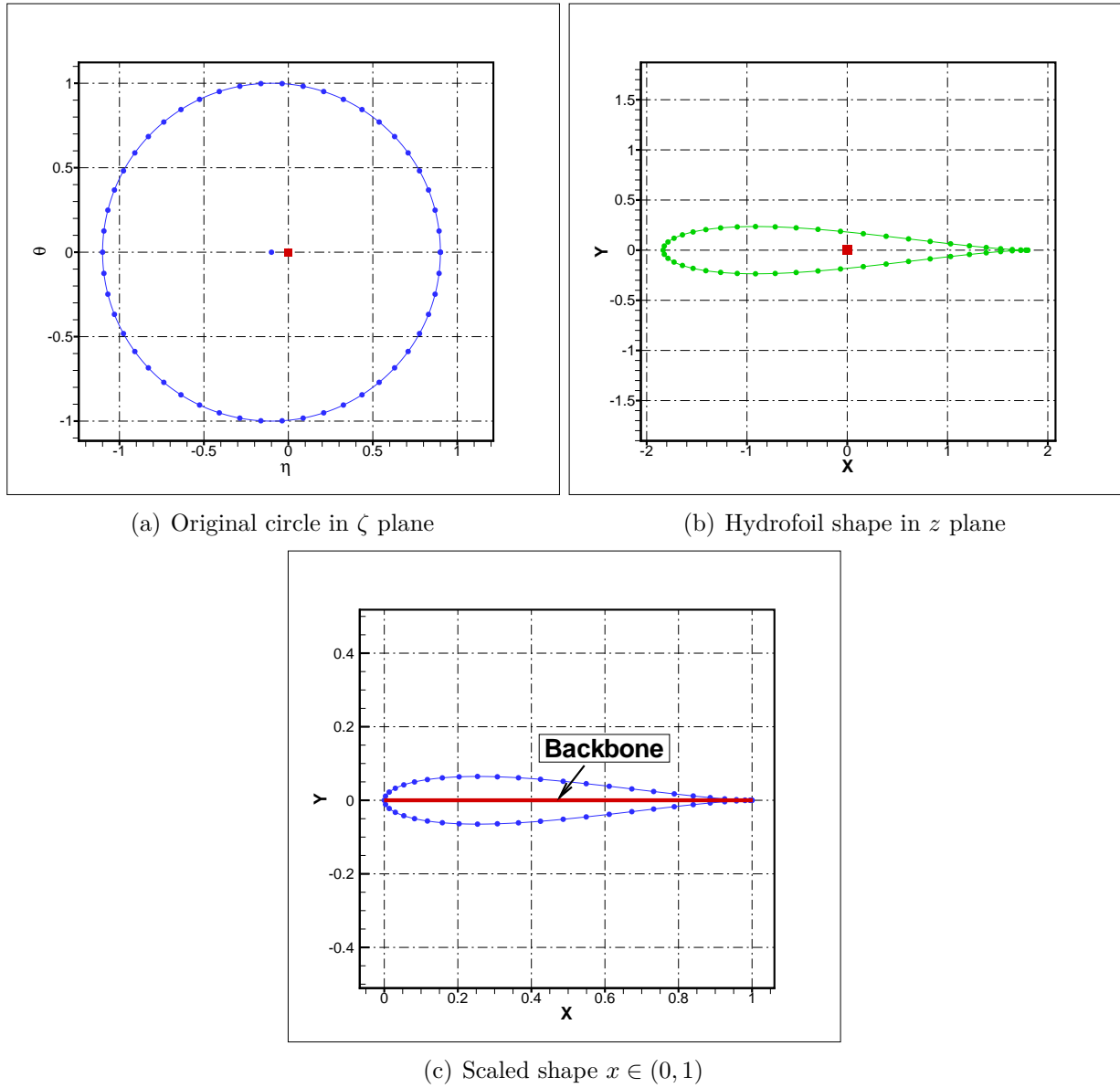


Figure 4.8: Sketch of the two-dimensional Joukowski transform. (a) The original circle in the ζ plane where $r_c = 1$, $\eta_c = -0.1$ and $\theta_c = 0$. (b) The hydrofoil shape in the z plane. (c) The shape is rescaled to fit $0 \leq x \leq 1$.

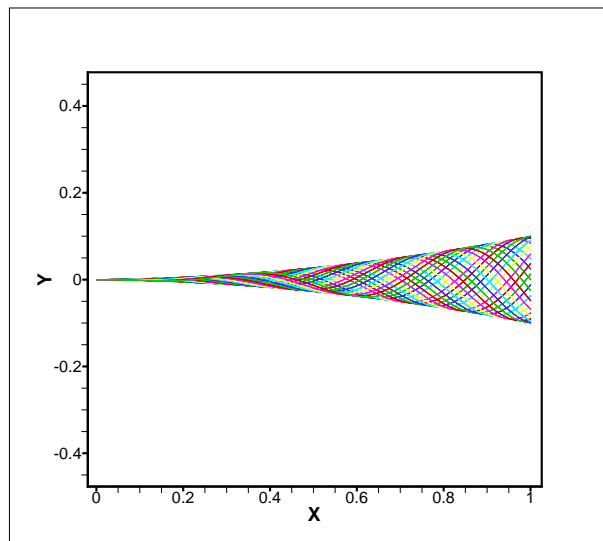


Figure 4.9: Backbone deformation according to Eq. (4.9) with $l = 1$ in one period, $\lambda = 0.5$, $a_0 = 0$, $a_1 = 0.01$ and $a_2 = 0.09$ to match the envelope used by Bergmann and Iollo [145].

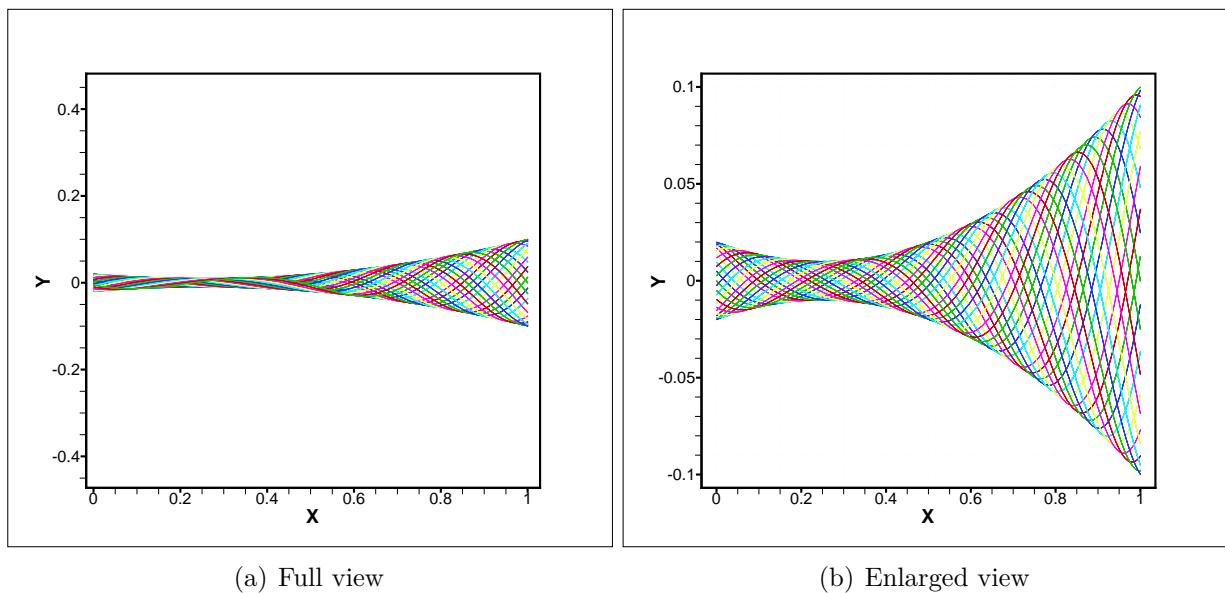


Figure 4.10: Backbone deformation according to Eq. (4.9) with $l = 1$ in one period, $\lambda = 0.5$, $a_0 = 0.02$, $a_1 = -0.08$ and $a_2 = 0.16$ to match the experimental envelope measured by Videler and Hess [41].

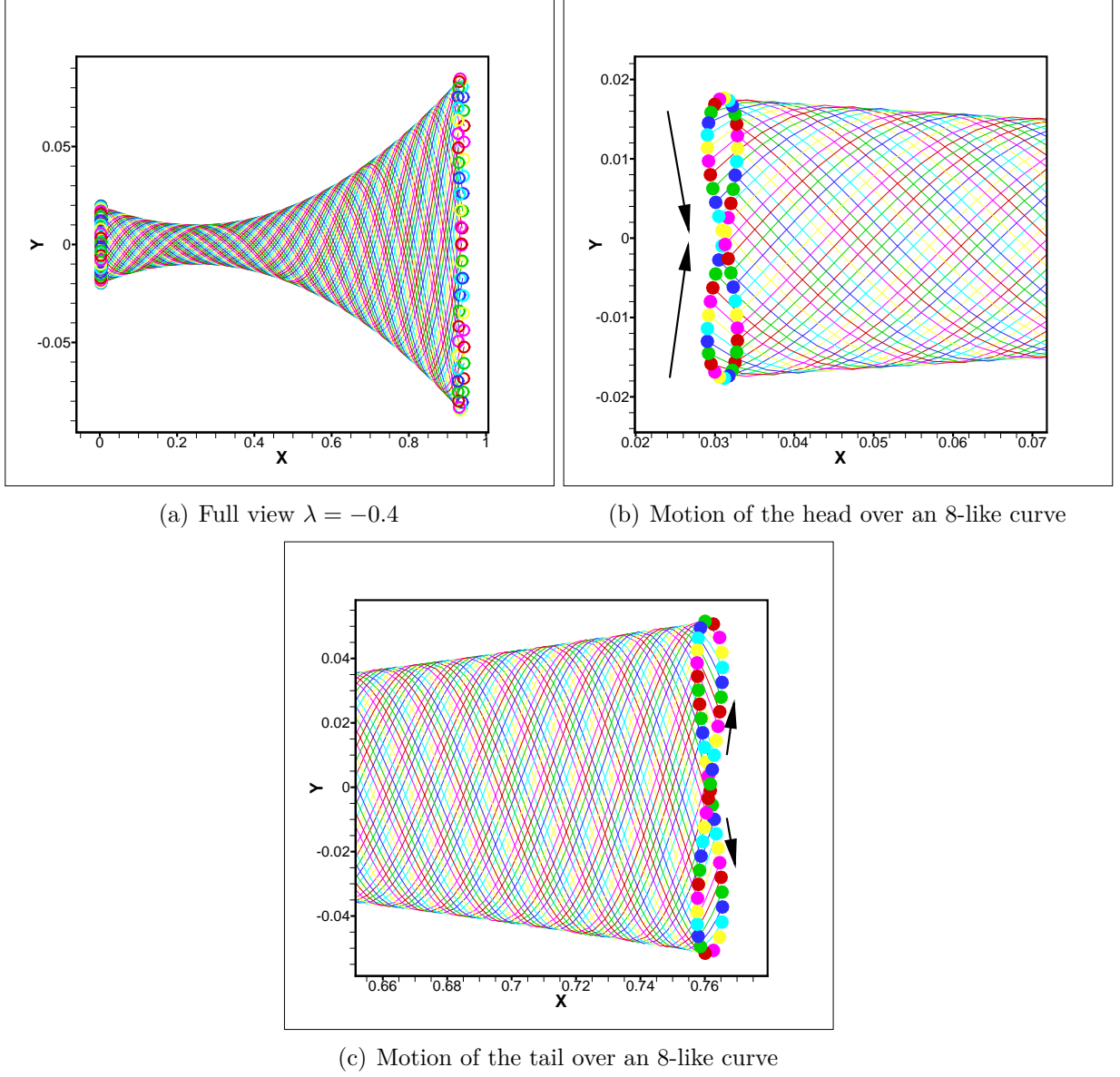
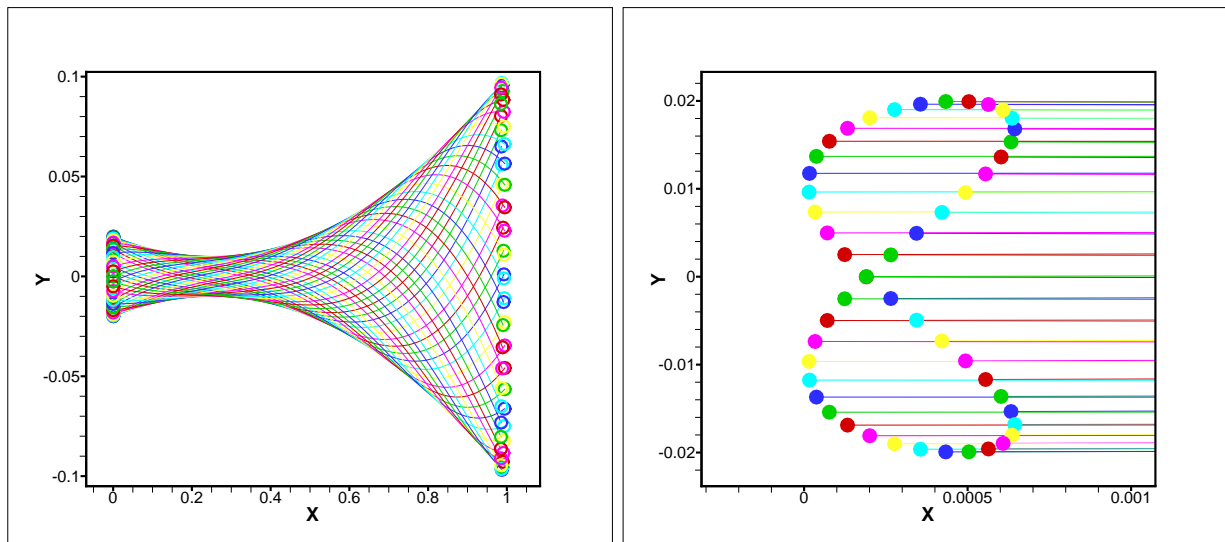
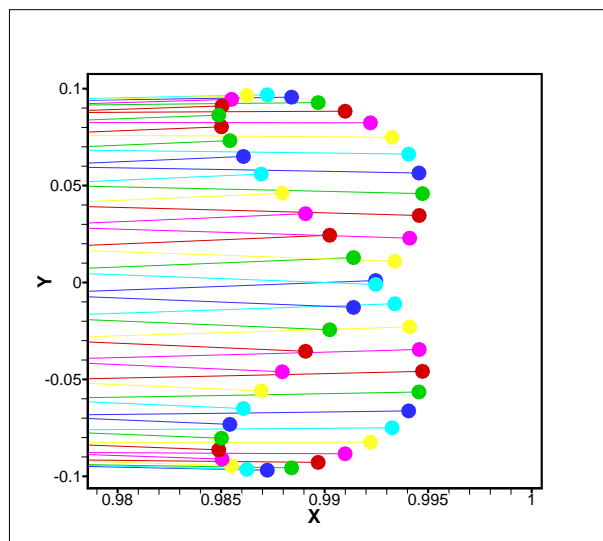


Figure 4.11: Backbone deformation according to Eq. (4.9) with $l = 1$ in one period, $\lambda = -0.1$, $a_0 = 0.02$, $a_1 = -0.08$ and $a_2 = 0.16$ to match the experimental envelope measured by Videler and Hess [41].



(a) Full view

(b) Motion of the head over an butterfly-like curve



(c) Motion of the tail over an butterfly-like curve

Figure 4.12: Backbone deformation according to Eq. (4.9) with $l = 1$ in one period, $\lambda = -1.5$, $a_0 = 0.02$, $a_1 = -0.08$ and $a_2 = 0.16$ to match the experimental envelope measured by Videler and Hess [41].

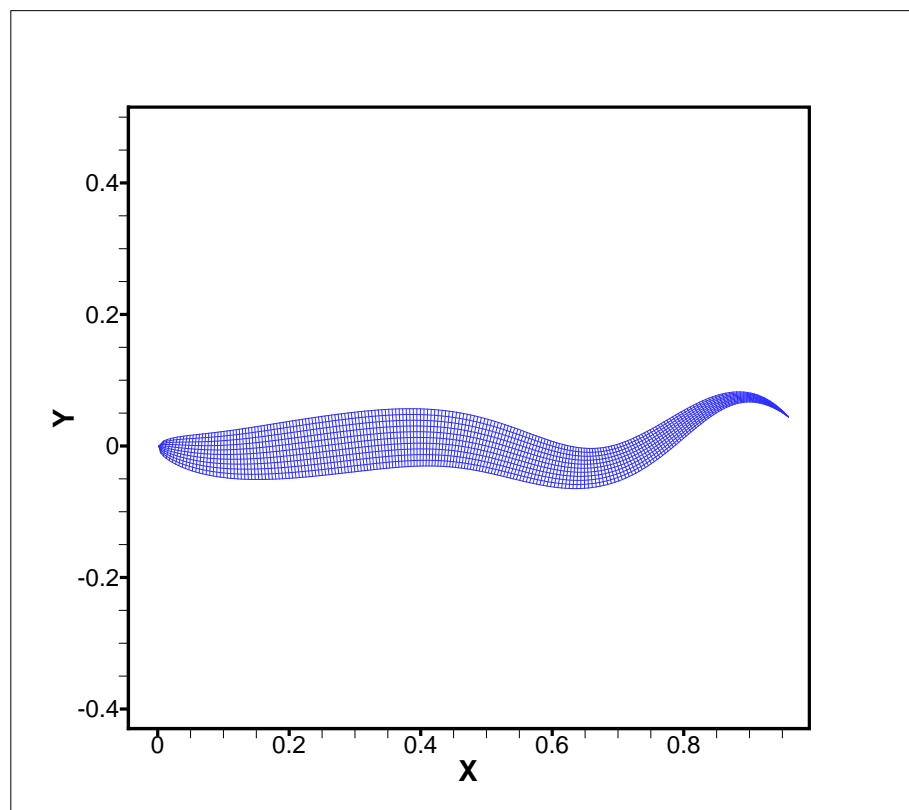
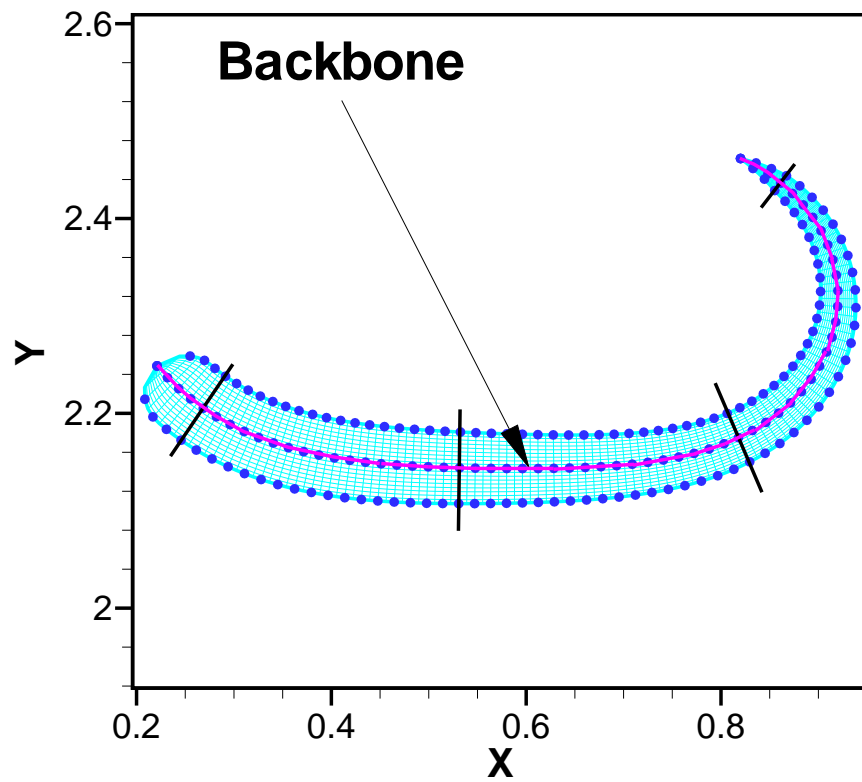


Figure 4.13: Body fitted structured grid for covering the fish. The mesh is generated by the normal to the backbone lines.

4.2.2 Kinematics of the fish based on curvature

The swimming mechanism in the majority of anguilliform and carangiform fishes can be modeled with a sinusoidal wave enveloped by a profile, lying over the backbone of the fish, which moves from the head to the tail. The geometrically exact theory of nonlinear beams, is developed by Simo [45] and extended for fish vertebral by Boyer et al. [106]. In this theory, the beam is considered as a continuous assembly of rigid sections of infinitesimal thickness, i.e., a one-dimensional Cosserat medium. We summarize the kinematics of the fish backbone in three dimensions for interested readers and future developments, but all the cases in this Chapter are limited to two dimensions. Following Boyer et al. [106], Rafei et al. [120] and Belkhiri [158] starting with the head orientation, position and velocities as boundary conditions, the kinematics of the backbone for anguilliform fishes can be determined by integration along the arclength $\xi \in [0, l_{\text{fish}}]$. The variation of the orientation along the backbone in terms of quaternions is obtained by

$$\frac{\partial Q}{\partial \xi} = \frac{1}{2} M^\vee(K) Q \quad , \quad \xi \in [0, l_{\text{fish}}] \quad (4.10)$$

where $Q = (\cos \frac{\phi}{2}, a_x \sin \frac{\phi}{2}, a_y \sin \frac{\phi}{2}, a_z \sin \frac{\phi}{2})^T$ are unit normalized quaternions, i.e., $(q_0^2 + q_1^2 + q_2^2 + q_3^2)^{1/2} = 1$, that represent the body frame orientation with respect to the inertial frame and $M^\vee(K)$ is an anti-symmetric tensor

$$M^\vee(K) = \begin{bmatrix} 0 & -k_1 & -k_2 & -k_3 \\ k_1 & 0 & k_3 & -k_2 \\ k_2 & -k_3 & 0 & k_1 \\ k_3 & k_2 & -k_1 & 0 \end{bmatrix} \quad (4.11)$$

where k_2 and k_3 in $K = (k_1, k_2, k_3)^T$ stand for the fish backbone transversal curvature and k_1 represents the rate of rotation (twist) of the section around the backbone with the normal aligned with the ξ -direction. The geometry $R = (x, y, z)^T$ in the Galilean reference frame is given by

$$\frac{\partial R}{\partial \xi} = \text{Rot}(Q) \Gamma \quad , \quad \xi \in [0, l_{\text{fish}}] \quad (4.12)$$

where $\Gamma = (\gamma_1, \gamma_2, \gamma_3)^T$ represents the local transversal shearing of the sections whose first component is the stretching rate along the ξ -direction. The rotation matrix in terms of the quaternions is then given by

$$\text{Rot} = 2 \begin{bmatrix} q_0^2 + q_1^2 - \frac{1}{2} & q_1 q_2 - q_0 q_3 & q_1 q_3 + q_0 q_2 \\ q_1 q_2 + q_0 q_3 & q_0^2 + q_2^2 - \frac{1}{2} & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_2 q_3 + q_0 q_1 & q_0^2 + q_3^2 - \frac{1}{2} \end{bmatrix} \quad (4.13)$$

The variations of mean linear, $V = (v_1, v_2, v_3)^T$, and angular, $\Omega = (\omega_1, \omega_2, \omega_3)^T$, velocities in the local frame, i.e., the frame attached to the body are given by

$$\frac{\partial}{\partial \xi} \begin{bmatrix} V \\ \Omega \end{bmatrix} = - \begin{bmatrix} K^\vee & \Gamma^\vee \\ 0 & K^\vee \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix} + \begin{bmatrix} \dot{V} \\ \dot{\Omega} \end{bmatrix}, \quad \xi \in [0, l_{\text{fish}}] \quad (4.14)$$

where superscript dot ($\dot{\cdot}$) represents the time derivative, (\vee) stands for the anti-symmetric matrix constructed from a given vector, e.g.,

$$K^\vee = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \quad (4.15)$$

The accelerations can also be deduced from the time derivative of Eq. (4.14). For more details we refer to Boyer et al. [106], Rafei et al. [120] and Belkhiri [158]. To find the velocities in the frame attached to the body from the velocities V_G in the Galilean reference frame and vice versa,

$$(v_1, v_2, v_3)^T = Rot^T(v_x, v_y, v_z)^T \quad (4.16)$$

can be used. By considering N ($1, \dots, N_{\text{points}}$) discrete points on the fish backbone, equations (4.10), (4.12) and (4.14) must be integrated simultaneously in space by a proper numerical method ($N_{eq} = 13$ in 3D). We are using a fourth-order Runge–Kutta method for integration and a comparison with a first-order Euler method shows that RK4 is more accurate, especially when the number of points along the fish backbone is less than $N_{\text{points}} = 30$.

4.2.3 Lagrangian structured grid covering the body

The first choice to start the parameterization of the swimmer body is a symmetric shape. The geometry of a two-dimensional swimmer can be characterized by the half width $w(\xi)$ of the body along its arclength (midline) $\xi \in [0, l_{\text{fish}}]$. Following the work of Kern and Koumoutsakos [105] and Carling et al. [67], the half width $w(\xi)$ is defined with Eq. (4.6). The shape of the fish before deformation is plotted in Fig. 4.14. In the mid part of the fish a linear function can also be used as in Gazzola et al. [144]. A structured grid formed by normal to backbone lines with thickness given by Eq. (4.6) covers the body. Some examples are shown in Figs. 4.13, 4.15 and 4.16. The velocity components of each point on the Lagrangian grid V_{shape} with indexes (I, J) , are given by

$$\vec{V}_{\text{shape}}(I, J) = \vec{V}_{\text{BN}}(I) + \vec{\Omega}_{\text{BN}}(I) \times \mathbf{r}(I, J) \quad (4.17)$$

where \vec{V}_{BN} and $\vec{\Omega}_{\text{BN}}$ are the linear and angular velocities of the backbone respectively, given by Eq. (4.14). The radius ($|\mathbf{r}| < w$) is measured over the transversal lines of the

structured grid normal to the backbone (see Fig. 4.13). Fig. 4.15 shows an example of the Lagrangian grid covering the fish after deformation in which the corresponding velocities of each point are also illustrated. Deformation of the body induces a divergent velocity field inside the body. Make sure to not add any artificial (linear and angular) momentum to the flow due to deformation of the body. In other words, in the absence of hydrodynamic forces and torque, displacement of gravity center and rotation around cg due to deformation (and thus linear and angular velocities) must be zero. We refer to Bergmann and Iollo [145] for more details.

The information of the Lagrangian structured grid covering the deformable body must be transferred to the Eulerian–Cartesian grid by interpolation to find $\chi(i, j)$ and $\mathbf{u}_p(i, j)$. To determine $\chi(i, j)$ on the Eulerian grid whose first point $(x, y)_{(1,1)} = (0, 0)$ is located at the origin, the coordinates of each point on the Lagrangian grid $\mathbf{X}_{\text{shape}}(I, J)$ are divided by Δx and Δy . After applying a correction to the integer part of the results they give the indexes (i, j) of the mask function χ on the Eulerian grid for which $\chi = 1$ is assigned. After determining the mask function $\chi(i, j)$ on the Eulerian grid, following Forestier [79], Minguez [122] and Kolomenskiy and Schneider [126] the mask is mollified by the Shuman [15] filter presented in Section 2.2.4

$$\bar{\chi}_{i,j} = (2\chi_{i,j} + \chi_{i+1,j} + \chi_{i-1,j} + \chi_{i,j+1} + \chi_{i,j-1})/6 \quad (4.18)$$

which is equivalent to a raised *cosine* filter in Fourier space, we refer to Pasquetti et al. [121] for more details. The effect of smoothing with Shuman filter is demonstrated in Fig. 2.11 for a one-dimensional box function $\chi(x) = 1, x \in [4.5, 5.5]$. The box function is represented by a red-solid line and the mollified box function $\bar{\chi}$ is plotted with a green-dashed line. An example of the transferred geometry χ to the Eulerian grid, after smoothing with Shuman filter is illustrated in Fig. 4.16, where the boundary of the Lagrangian grid is also added to the contours of the smoothed mask function $\bar{\chi}$. It can be seen that it lies between the maximum and minimum values of the mask function. Smoothing of the mask function $\bar{\chi}$ reduces the stiffness of the vorticity transport equation, thus larger time steps can be used. It also increases the regularity of the pressure and the velocity field. Moreover, in dealing with moving boundaries, when the mask functions is smooth the oscillations of the hydrodynamic coefficients are weaker. However, even without filtering of the mask function, by applying a central second-order finite difference method the solution converges. But by using fourth and higher order discretizations smoothing of the mask function becomes necessary, if not Gibbs oscillations or divergence is expected, especially when the mask function χ is moving. Note also that the interpolated velocity field \mathbf{u}_p on the Eulerian grid is not divergence-free, we refer to Gazzola et al. [144] for a complete theoretical and numerical discussion about this subject. In the present investigation we do not consider this issue under the assumption that the body is slender. We use two-

dimensional linear interpolation, to transfer the velocities of the Lagrangian grid given by Eq. (4.17) to the Eulerian grid. By considering

$$f(x, y) = axy + bx + cy + d \quad (4.19)$$

and using the four nearest points of the Lagrangian grid, Eq. (4.19) leads to a 4×4 linear system for each point with $\bar{\chi} \neq 0$ on the Eulerian grid. To determine the unknowns, the system is solved by a direct method, i.e., Gauss-Jordan elimination from [56]. For all points in the interior of the fish we have $\chi(i, j) = 1$ on the Eulerian grid. For each point of the Eulerian grid in which $\bar{\chi} = 1$ the four nearest points of the Lagrangian grid are used to find the coefficients of the linear system formed by (4.19). In some points of the Eulerian grid due to mollifying the mask function χ by Eq. (2.52) we have $0 < \bar{\chi} < 1$, therefore the interpolation automatically becomes an extrapolation. Some points are completely outside of the original Lagrangian shape. At the start and the end singularities of the Lagrangian grid, where points are repetitive, just one of the points can be used for finding the penalized velocities \mathbf{u}_p over the Eulerian grid, if not the interpolation matrix will have a zero determinant (singular). However, the start and the end points are used for determining the mask function. An example of the interpolated velocity components on the Eulerian grid is illustrated in Fig. 4.16.

The spacing of the grid points on the Lagrangian grid $\Delta\mathbf{X}$ must be fine enough in comparison to Δx and Δy to represent accurately the deformation of the body on the Eulerian grid, i.e., $\Delta\mathbf{X} \leq \Delta\mathbf{x}$. However, the ratio $\Delta\mathbf{x}/\Delta\mathbf{X}$ cannot be determined exactly because $\Delta\mathbf{X}$ is varying even if Δx and Δy are fixed. Nevertheless in Figs. 4.15 and 4.16 the Lagrangian and the Eulerian grids are schematically illustrated for a fine and coarse Lagrangian grid. If the Lagrangian grid is very fine, the computational effort in the procedure of evolving the mask function χ and determining the corresponding velocities \mathbf{u}_p will increase. The additional cost does not lead to considerable enhancement in the accuracy of the mask function χ or the interpolated velocities of the body \mathbf{u}_p on the Eulerian grid. However, a very fine Lagrangian grid may lead to singular matrices in the interpolation procedure via Eq. (4.19) because the four points chosen for interpolation will be very close. For a very fine Lagrangian grid zero order interpolation must be used, i.e., the velocities of the nearest point on the Lagrangian grid must be assigned to the corresponding Eulerian grid. On the other hand if a very coarse Lagrangian grid is used the information of the body will be lost. Especially the rotational velocity field due to the deformation of the body which has a great importance in the accuracy of the simulations, will be missed and even divergence of the simulations is expected. Moreover, the values of the mask function will not reach the value one inside the fish with insufficient resolution of the Lagrangian grid, see Fig. 4.16. The geometry will not be accurate near singular points (like the tail) or boundaries with high curvature (like the head). The hydrodynamic coefficients can also be inaccurate

whenever a coarse grid is used for the Lagrangian grid. An optimal value in the sense of accuracy and computational effort is proposed for the size of the Lagrangian grid

$$\frac{\Delta \mathbf{x}}{10} < \Delta \mathbf{X} < \frac{\Delta \mathbf{x}}{2}$$

A summary of the algorithm for the fluid interaction with a deformable body is given in

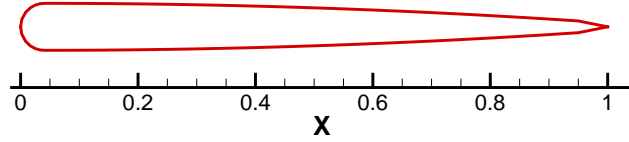


Figure 4.14: Shape of the fish given by Eq. (4.6) before deformation.

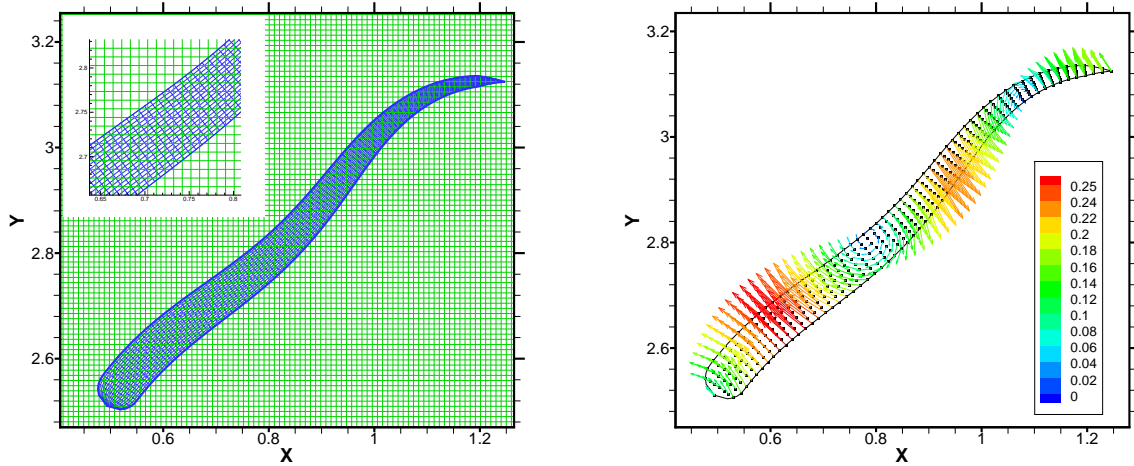


Figure 4.15: Left: The Lagrangian structured grid ($Im_b \times Jm_b = 121 \times 19$) over the Eulerian one. Right: The Lagrangian structured grid covering the fish after deformation and the corresponding velocity vectors of each point, colored by absolute velocity $\sqrt{u^2 + v^2}$.

Algorithm 2. The flowchart is illustrated in Fig. 2.18.

4.3 Validation of the algorithm for deformable bodies

The anguilliform swimming presented in Gazzola et al. [144] is considered for validation of the proposed algorithm to deal with deformable bodies interacting with incompressible flows. The considered test case is a swimming fish in a forward gait. A periodic swimming law is defined by fitting the backbone of the fish to a given curve $y(x, t)$ while keeping the backbone length l_{fish} fixed. Let ξ be the arclength over the curvilinear coordinate

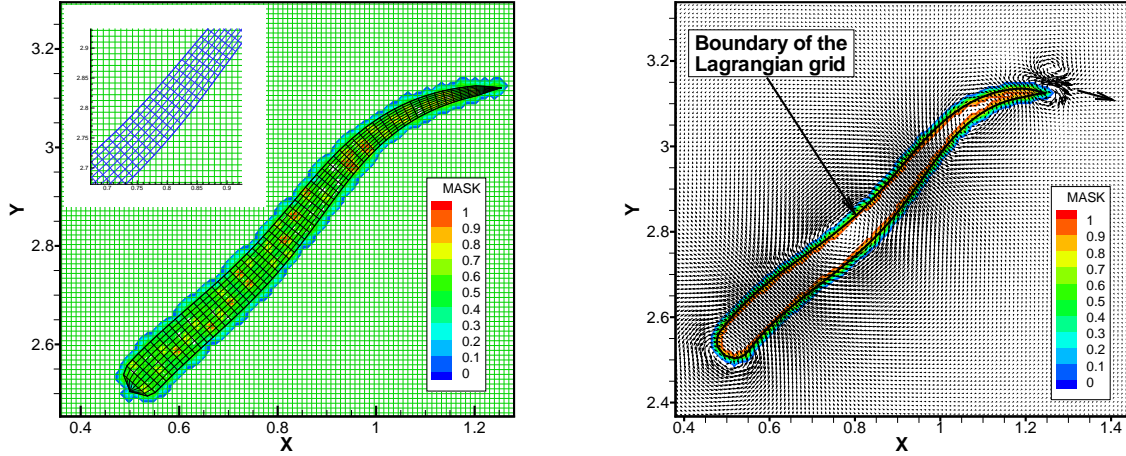


Figure 4.16: Left: Interpolated mask function $\bar{\chi}$ from a Lagrangian grid ($Im_b \times Jm_b = 61 \times 9$) where $\Delta \mathbf{x} < \Delta \mathbf{X}$, as can be seen insufficient resolution in Lagrangian grid results in $\bar{\chi} = 1$ in very few points inside the fish. Right: Boundary of the Lagrangian grid (black line) and the interpolated smoothed mask function $\bar{\chi}$ (colored isolines). As can be seen the boundary of the Lagrangian grid ($Im_b \times Jm_b = 121 \times 19$) lies between minimum and maximum values of the mask function. The velocity components forming a jet at the tail while the fish is turning.

of the deformed backbone ($0 \leq \xi \leq l_{\text{fish}}$). For points being uniformly distributed with $\Delta \xi = l_{\text{fish}}/(N - 1)$ over the backbone, y is given by

$$y(x, t) = a(x) \sin(2\pi(x/\lambda + ft)) \quad (4.20)$$

where λ is the wavelength of the imposed deformation, f represents the frequency of the backbone undulation and the envelope $a(x)$ is given by

$$a(x) = a_0 + a_1 x + a_2 x^2 \quad (4.21)$$

where x is defined by inverting the arclength integral, i.e., $\Delta x = \Delta \xi / \sqrt{1 + (\partial y / \partial x)^2}$. The wavelength of the fish is defined in accordance with the geometry of the backbone in the Cartesian coordinate. The pointwise curvature of the backbone is needed to use the geometrically exact theory of nonlinear beams, described in Section 4.2.2. One must switch from the Cartesian system to the curvature, thus the second derivative of Eq. (4.20) gives us the propulsion curvature as follows:

$$k_{\text{prop}}(\xi, t) = (2a_2 - (2\pi/\lambda)^2 a(\xi)) \sin(2\pi(\xi/\lambda + ft)) \\ + (4\pi(a_1 + 2a_2\xi)/\lambda) \cos(2\pi(\xi/\lambda + ft)) \quad (4.22)$$

where $a(\xi) = a_0 + a_1\xi + a_2\xi^2$. Using the curvature of the backbone provides a general framework which is independent of the Cartesian coordinates (direction), this is especially interesting to prevent the ambiguity in definition of the geometry when the fish performs a complete rotation. The parameters used by Kern and Koumoutsakos [105] and Gazzola et al. [144] for the kinematics of the fish are as follows; $\lambda = 1$, $f = 1$, $a_2 = 0$, $a_1 = 0.125/(1+c)$, $a_0 = 0.125c/(1+c)$ and $c = 0.03125$. The profile of the fish is given by Eq. (4.6) and is plotted in Fig. 4.14. The buoyancy is equal to zero, i.e., $\rho_b = \rho_f$. The viscosity of the fluid is set to $\nu = 1.4 \times 10^{-4}$ resulting in an approximative Reynolds number $Re \approx 3800$, with an asymptotic mean velocity $U_{\text{forward}} \approx 0.52$.

The simulations of Gazzola et al. [144] are carried out on a rectangular domain $(x, y) \in [0, 8l_{\text{fish}}] \times [0, 4l_{\text{fish}}]$ with resolution 4096×2048 and a penalization parameter equal to $\eta = 10^{-4}$. We perform the simulations on a rectangular domain $(x, y) \in [0, 10l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$ by imposing a penalization parameter inside the body equal to $\eta = 10^{-3}$ with resolutions of 2048×1024 and 1024×512 and $\Delta t = 10^{-3}$. The centroid of the fish is initially positioned at $x_{\text{cg}} = 0.9L_x$ and $y_{\text{cg}} = 0.5L_y$. Two snapshots of vorticity isolines at $t = 1$ and $t = 9$ with the aforementioned parameters are illustrated in Fig. 4.19. The forward velocities corresponding to gravity center (cg) of the fish, computed with different methods and parameters are compared with those of Kern and Koumoutsakos [105] and Gazzola et al. [144] in Fig. 4.18. We impose two degrees of freedom fixing the angular velocity of the fish around center of gravity equal to zero. But this does not result in a motion without slaloming. Deformation of the fish in addition to the lateral displacement creates slaloming. The simulations start with the body $\mathbf{u}_P(\mathbf{x}, 0) = 0$ and fluid at rest, i.e., $\omega(\mathbf{x}, 0) = \psi(\mathbf{x}, 0) = 0$. Free-slip boundary conditions are imposed at the four surrounding walls ($\psi|_{\partial\Omega} = \omega|_{\partial\Omega} = 0$). In the reference simulations of Kern and Koumoutsakos [105] and Gazzola et al. [144] the motion of the fish is initialized by gradually increasing the amplitude of the backbone through a sinusoidal function (plotted in Fig. 4.17), from zero to its designated value during the first period T . Here we do not consider this and start by a sudden movement given by Eq. (4.20). That is the reason why a deviation from the reference solution can be seen in the first period. This deviation will continue systematically until the asymptotic velocity is reached at $t = 7$.

The reference simulation of Kern and Koumoutsakos [105] is based on a body fitted grid with a finite volume discretization which is first-order in time and second-order in space. The Navier–Stokes equations were solved using the commercial package STAR-CD which uses arbitrary Lagrangian–Eulerian grids. The solution of Newton’s equations of motion and the deformation and displacement of the Lagrangian grid are implemented in user defined subroutines linked to STAR-CD. The implemented explicit coupling procedure is a staggered integration algorithm proposed by Farhat and Lesoinne [78]. The simulation of Gazzola et al. [144] is based on a remeshed vortex particle code coupled with Brinkman penalization which handles arbitrarily deforming bodies and especially the divergent ve-

locity field inside the body is considered. A projection method is used by Gazzola et al. [144]. The resulting Poisson equations for rotational (solenoidal) and potential (divergent) components of the velocity fields are solved in an unbounded domain, based on a method, using FFT on a Cartesian grid. A second-order finite difference discretization in two dimensions and a fourth-order finite difference discretization in three dimensions are used for all spatial derivatives. The time step is adapted by a Lagrangian CFL condition. The difference on the final forward velocity of the fish, reported by Gazzola et al. [144] by taking into account the divergence of the velocity field inside the fish due to deformation, is visible in Fig. 4.18. Even though the average divergence over the fish volume is zero (i.e. the volume is conserved), locally inside the fish the velocity field is not divergence free. We do not deal with this issue in the present study under the assumption that the body is slender. In our simulations a grid independent solution is obtained with 2048×1024 grid points. The difference of the forward velocity in two simulations with 2048×1024 and 1024×512 grid points can be seen in Fig. 4.18. Filtering of the hydrodynamic coefficients is necessary to prevent the simulation from divergence and non-physical results. We are using a second-order exponential filtering (2.25) instead of the first-order filtering used by Kern and Koumoutsakos [105] (see the discussions in Sections 2.1.2 and 3.4.2). This process is like adding a damper to the system, therefore a proper value for δ must be chosen via numerical tests to obtain reliable and physical results. We propose values in the range of $\delta \in [0.0001, 0.01]$ for fluid–solid interaction problems. However this can also depend on the manner of non-dimensionalization of the forces. In Fig. 4.18 the effect of filtering with two filter parameters, i.e., $\delta = 0.001$ and $\delta = 0.05$, can be seen. The simulations with a smaller filter parameter, e.g., $\delta = 0.001$, are more stable but instead will lead to smaller amplitudes in the oscillations of the terminal velocity. A sensitivity analysis is thus necessary.

4.4 Application and results

In this section we attempt to propose an efficient law for rotation control of an anguilliform swimmer. Fish maneuvering to attain a predefined fixed goal is done by adding a constant curvature $k_{\text{offset}}(t)$ all along the fish backbone $\xi \in [0, l_{\text{fish}}]$, to the primary propulsion mode:

$$k_3 = k_{\text{prop}}(\xi, t) + k_{\text{offset}}(t) \quad (4.23)$$

For the fish in forward gait k_{offset} is set equal to zero. To perform a rotation, a desired curvature k_{des} must be evaluated by the following relation,

$$k_{\text{des}}(\theta_{\text{des}}) = \begin{cases} -\text{sgn}(\theta_{\text{des}}) k_{\text{max}} & |\theta_{\text{des}}| \geq \theta_{\text{limit}} \\ -\text{sgn}(\theta_{\text{des}}) k_{\text{max}} \left(\frac{\theta_{\text{des}}}{\theta_{\text{limit}}}\right)^2 & \text{elsewhere} \end{cases} \quad (4.24)$$

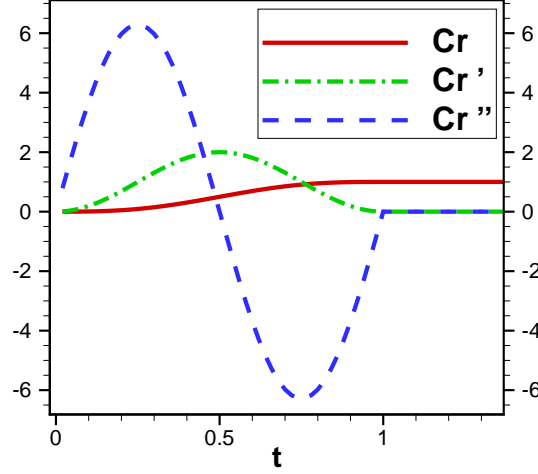


Figure 4.17: Smooth step function proposed by Boyer et al. [106] for gradually evolving the fish curvature in the first stroke $Cr(t) = t' - \sin(2\pi t')/(2\pi)$, $t \in [t_i, t_f]$ with $t' = (t - t_i)/(t_f - t_i)$, $t_i = 0$, $t_f = 1$. At $t = 1$ the left- and right-hand limits are equal for the function Cr and its first Cr' and second Cr'' derivatives.

where sgn represents the sign function, i.e., $sgn(\theta_{des}) = \theta_{des}/|\theta_{des}|$, positive and negative values of θ_{des} in the head frame will push the fish to turn left and right, respectively. For a schematic representation of θ_{des} see Fig. 4.20. The algorithm of rotation control consists of following steps; At each time step first by considering the desired goal, according to the position and direction of the head, a desired angle θ_{des} should be calculated. Then using Eq. (4.24) a desired curvature k_{des} must be found. After that k_{offset} will be evaluated with the following relation,

$$k_{offset}^{n+1}(k_{des}) = \begin{cases} k_{offset}^n + \Delta k & k_{offset}^n < k_{desired} \\ k_{offset}^n - \Delta k & k_{offset}^n > k_{desired} \\ k_{offset}^n & k_{offset}^n = k_{desired} \end{cases} \quad (4.25)$$

where $\Delta k = \Delta t \pi / T$. Then k_{offset} must be added to the primary propulsion curvature for performing a rotation. However, the change of the added curvature k_{offset} given by Eq. (4.25) must be gradually, i.e., $O(\Delta t)$ to perform a physically reasonable rotation. Finally, knowing the direction, the position and the velocities of the head, equations (4.10), (4.12) and (4.14) must be integrated simultaneously in space to obtain the position and the velocities of the backbone. In the case of anguilliform swimming the length of the fish is constant, we have a stretching rate equal to one over the backbone of the fish and the local transversal shearing is equal to zero, therefore $\Gamma = (1, 0, 0)$ is used in Eq. (4.12). In two-dimensional swimming just one curvature can be imposed, i.e., $K = (0, 0, k_3)$, the twist and transversal curvature are equal to zero. By considering $Im_b = 251$ discrete points on the backbone of the fish and $Jm_b = 39$ points in the lateral direction, a Lagrangian structured

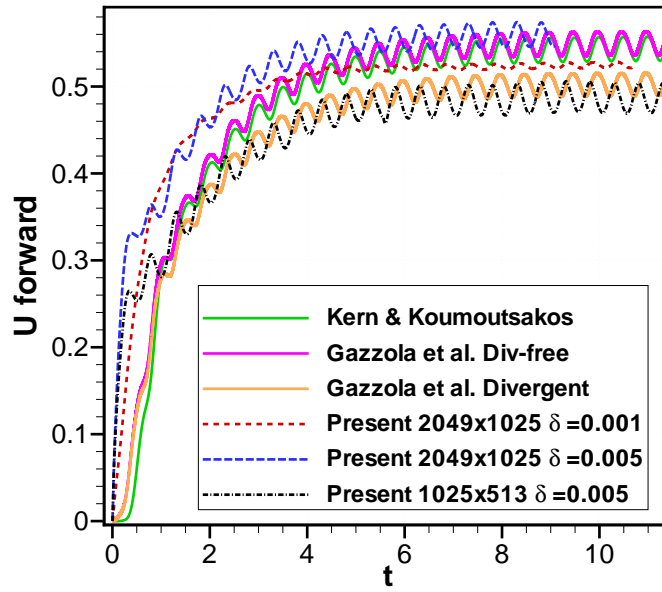


Figure 4.18: Forward velocity U of a 2D anguilliform swimmer ($\lambda = f = 1$). Solid lines indicate the reference simulations performed by Kern and Koumoutsakos (green) [105] and Gazzola et al. (pink and brown) [144]. Dashed lines represent the results with the proposed algorithm.

grid is constructed which covers the fish. By choosing $k_{\max} = \pi$ in Eq. (4.24) the fish lies over a semicircle when it turns with its maximum curvature. As in Bergmann and Iollo [145] we are using $\theta_{\text{limit}} = \pi/4$. The time derivative of the curvature dk/dt is needed in Eq. (4.14) for evaluation of the velocity and can be determined numerically. A simulation is performed to show the performance of the proposed law for rotation control of a swimmer toward a predefined goal. The domain size is $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, the resolution is set to 1024×1024 , the penalization parameter $\eta = 10^{-3}$, the filter parameter for denoising the hydrodynamic coefficients $\delta = 0.005$, tail beat frequency $f = 1$ and the wavelength of deformation is $\lambda = 1$. The backbone envelop is parametrized with $a_2 = 0$, $a_1 = 0.125/(1+c)$, $a_0 = 0.125c/(1+c)$ and $c = 0.03125$. The profile of the fish is given by Eq. (4.6) and plotted in Fig. 4.14. The kinematic viscosity is $\nu = 1.4 \times 10^{-4} \text{ m}^2/\text{s}$, the initial position of the head $(x_0, y_0) = (0.1L_x, 0.5L_y)$ and the initial angle of the head is $\theta_0 = 0$. Fig. 4.21 shows snapshots of vorticity isolines, obtained during a simulation of the fish swimming toward a predefined goal which is located at $(x_f, y_f) = (0.9L_x, 0.5L_y)$. The simulations start with the body, $\mathbf{u}_P(\mathbf{x}, 0) = 0$, and surrounding fluid at rest, $\mathbf{u}(\mathbf{x}, 0) = 0$, i.e., $\omega(\mathbf{x}, 0) = \psi(\mathbf{x}, 0) = 0$. Free-slip boundary conditions are imposed at the four surrounding walls ($\psi|_{\partial\Omega} = \omega|_{\partial\Omega} = 0$). The motion of the fish is initialized by gradually increasing the curvature of the backbone, given by Eq. (4.23), through a sinusoidal function (plotted in Fig. 4.17), from zero to its designated value during the first period T . After reaching the vicinity ($r_{\text{goal}} = 0.5l_{\text{fish}}$) of

the goal, the curvature of the backbone, given by Eq. (4.23), will tend to zero (see Fig. 4.22) by multiplying it with the following function

$$C(t) = \frac{t_f - t}{t_f - t_i} + \frac{1}{2\pi} \sin\left(2\pi \frac{t - t_i}{t_f - t_i}\right), \quad t \in [t_i, t_f] \quad (4.26)$$

which is the mirror of the function presented in Fig. 4.17, with $t_i = t_{\text{reached}}$, $t_f = t_{\text{reached}} + T$ for gradually decreasing the curvature of the backbone during one period. Samples of the backbone of the fish are plotted in Fig. 4.25. As can be seen in Fig. 4.21 the values of the vorticity start from zero and go up very fast during the rotation, i.e., $\omega \in [-200, 220]$. In the forward gait the range of the vorticity is $\omega \in [-60, 70]$ and finally it goes down by stopping the stroke in the vicinity of the goal to be in the range of $\omega \in [-28, 25]$. Saddle and center points in the flow separated from the fish can successively be seen in Figs. 4.23 and 4.24. These are the common characteristics of separated flows. The pressure is not present in the algorithm and is just computed for visualization purpose. For evaluation of the pressure field the Poisson equation (2.20) can be simplified ($\rho_f = 1$) for the current application as follows:

$$\nabla^2 p = 2(u_x v_y - u_y v_x) - \nabla \cdot [\eta^{-1} \chi(\mathbf{u} - \mathbf{u}_p)] \quad (4.27)$$

where Neumann boundary conditions, $\partial p / \partial \mathbf{n}|_{\partial\Omega} = 0$, are imposed at the borders of the rectangular domain. Using a second-order forward finite difference discretization one has

$$p_1 = (4p_2 - p_3)/3$$

at the left boundary. Similar backward/forward relations can be derived for right, up and down boundaries. A point successive over relaxation (PSOR) method [56] with red-black sweeping is used for computing the pressure field once every 500 iterations. During the iterations, the value of the pressure in the center of the cavity is fixed $p_{\text{center}} = \text{cte}$, i.e.,

$$p(N_x/2, N_y/2) = 1$$

to avoid the singularity in solution of the Poisson equation due to imposed Neumann boundary conditions at the borders. Snapshots of the pressure isolines are illustrated in Fig. 4.22. High and low pressure regions on the either side of the fish can be seen. As expected the pressure contours are normal to the boundary of the fish and the boundaries of the computational domain. The centers of the vortices correspond to low pressure regions. The deviation of the pressure from $p_\infty = 1$ goes up to $p \in [-21, 27]$ after starting the straight swimming at $t = 5$ and goes down instantaneously when the fish reaches the vicinity of the goal ($t = 15$), thus stopping the stroke. This is in clear contradiction with the vorticity field which is very persistent even after stopping the stroke and shows the global nature of the

pressure field against the localized nature of the vorticity field. A high pressure region is seen between the head and the tail of the fish at $t = 2.25$ when it turns with the maximum curvature $k = \pi$ forming a c-shape which corresponds to what is observed by Gazzola et al. [155]. The C-bent maneuver before fish escape, has been explained by Gazzola et al. [155] to be effective in trapping and accelerating larger volumes of fluid. Despite the solid objects facing a free-stream, in which the maximum pressure occurs at the stagnation point of the front, in the swimming fish the high and low pressure regions occur on either side of the fish alternatively. However at the final stage of the motion after stopping the stroke, a high-pressure region at the head of the fish is observed at $t = 15$, which is illustrated in Fig. 4.22 (l). The smoothing of the mask function $\bar{\chi}$ by Eq. (2.52) results in a smooth pressure field, there are no oscillations inside and around of the fish and the pressure distribution is regular. With the proposed law for rotation of the fish which adds a time-dependent curvature (constant all along the backbone) to the primary propulsion curvature, starting from rest the fish executes a sharp 180° turn within an area of about 1.3 times its body length.

4.5 Conclusion

In this chapter the algorithm for simulation of deformable bodies interacting with two dimensional incompressible flows is presented and examined. The simulation of the fish in forward gait is considered for validation of the proposed algorithm by comparing the results with those of Gazzola et al. [144]. Even if the spatial resolution in our simulations is half of that used by Gazzola et al. the results are in good agreement. This shows the advantage of the structured grid used for description of the fish shape in our simulations in comparison to immersed grid used by Gazzola et al. [144]. Proper denoising of the hydrodynamic coefficients is crucial in dealing with fluid–solid interaction problems via the volume penalization method. An efficient law for curvature control of an anguilliform swimmer toward a predefined goal is proposed which is based on the geometrically exact theory of nonlinear beams. With the proposed law the motionless fish executes a sharp 180° turn within an area of about 1.3 times its body length. Validation of the developed method shows the efficiency and expected accuracy of the algorithm for rotation control of an anguilliform swimmer. Some perspectives for future works are adding a multiresolution analysis to the algorithm for grid adaptation, enhancement of rotation control law and extension of the algorithm to three dimensions.

Algorithm 2 Fluid interaction with a deformable body

1. START FROM AN INITIAL CONDITION
 2. BODY KINEMATICS
 - (a) (Just for the fish) Create Eel's backbone by integrating Eqs (4.10), (4.12) and (4.14)
 - (b) (Just for the fish) Cover the shape by a Lagrangian structured grid & compute velocities of each point with Eq. (4.17). Make sure to not add any artificial (linear and angular) momentum to the flow.
 - (c) Compute the mask $\chi(i, j)$ and smooth it by Eq. (2.52)
 - (d) Compute the moment of inertia J around the reference point
 - (e) Compute the velocity components of the body $u_p(i, j), v_p(i, j)$ on the Eulerian grid by interpolation (Lagrange \rightarrow Euler)
 3. TIME INTEGRATION OF THE FLOW FIELD WITH RK4
 - (a) $\omega_0 = \omega^n$, $\psi_0 = \psi^n$
For $i = 1, 2, 3$ ($\alpha_1 = 1/2$, $\alpha_2 = 1/2$ and $\alpha_3 = 1$)
 - (b) Compute $k_i(\omega, \psi)^{i-1}$ from Eq. (2.88)
 - (c) $\omega_i = \omega^n + \alpha_i \Delta t k_i$
 - (d) Solve Eq. (2.6); $-\nabla^2 \psi_i = \omega_i$ for updating (u, v)
End For
 - (e) Compute $k_4(\omega_3, \psi_3)$ from Eq. (2.88)
 - (f) Update vorticity from Eq. (2.87); $\omega^{n+1} = \omega^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
 - (g) Solve Eq. (2.6); $-\nabla^2 \psi^{n+1} = \omega^{n+1}$
 4. SOLVE FOR THE BODY DYNAMICS
 - (a) Compute the hydrodynamic coefficients of the body from Eqs. (2.22) and (2.23)
 - (b) Denoise the coefficients by Eq. (2.25)
 - (c) Compute the displacements from Eq. (2.28)
 - (d) Compute the rotation from Eq. (2.30)
 5. WRITE NECESSARY DATA TO FILE
 6. IF $T < T_{end}$, GO TO STEP 2
 7. END
-

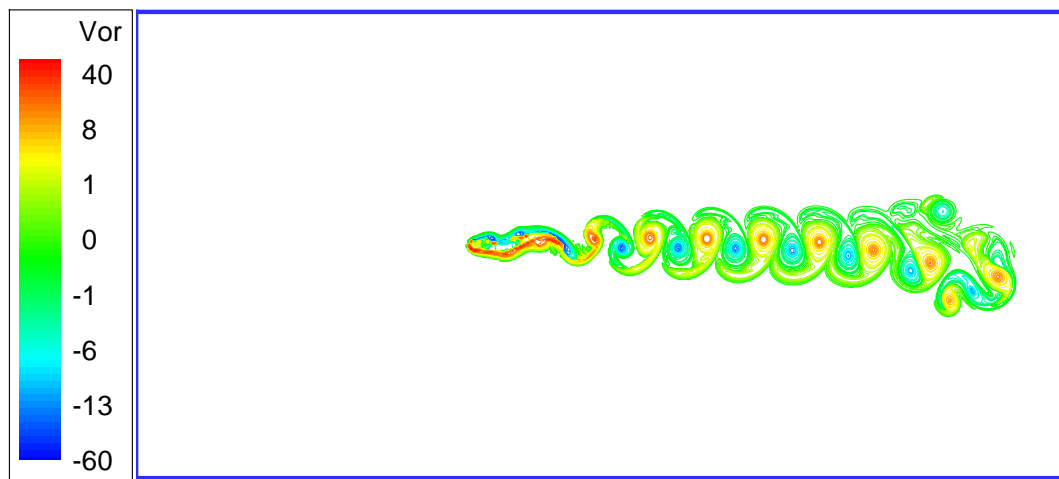
(a) $t = 1$ (b) $t = 9$

Figure 4.19: Snapshots of vorticity isolines obtained during a simulation in a rectangular domain $(x, y) \in [0, 10l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, with resolution 2048×1024 , by imposing a penalization parameter inside the body equal to $\eta = 10^{-3}$ and the time step $\Delta t = 10^{-3}$. The filter parameter for denoising of the hydrodynamic forces is $\delta_{\text{filter}} = 0.005$ and the Reynolds number is approximately $Re \approx 3800$.

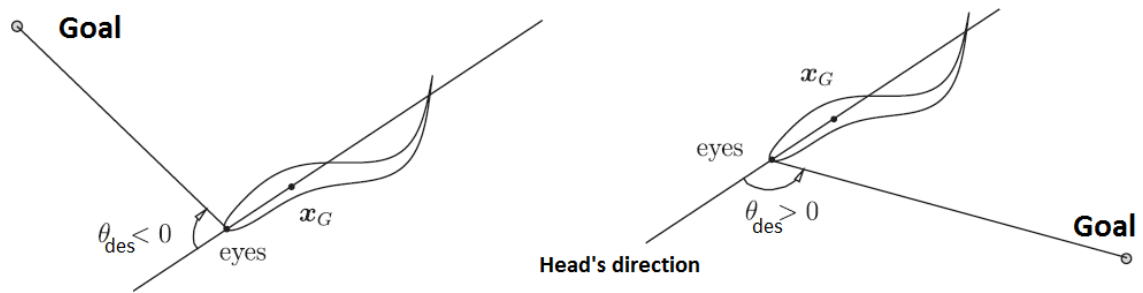


Figure 4.20: Schematic representation of the desired angle for curvature control, during the rotation of the fish toward the goal. Here $\theta_{des} = \theta_{goal} - \theta_{Head}$ is the difference of the angles between head direction and the line passing through the target and the head ($-\pi < \theta_{des} < \pi$), picture adapted from Bergmann and Iollo [145] with a slight modification.

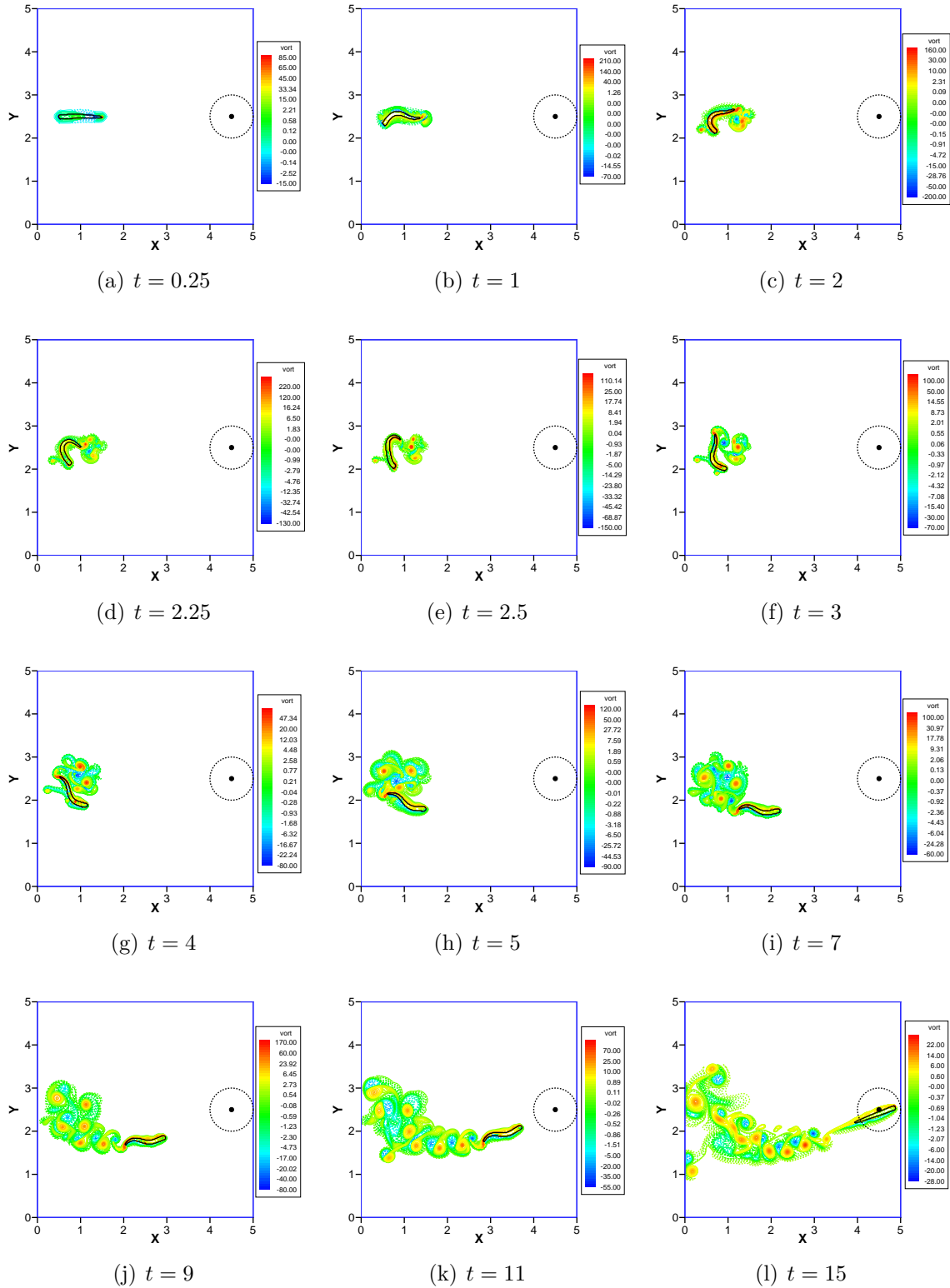


Figure 4.21: Snapshots of vorticity isolines obtained during a simulation of swimming fish toward a predefined target which is located at $(x_f, y_f) = (0.9L_x, 0.5L_y)$. At $t = 0$ the fish and the surrounding flow are at rest. After reaching the vicinity ($r = 0.5l_{\text{fish}}$) of the target the curvature of the backbone tends to zero by Eq. (4.26). The domain of the solution is $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, the resolution of the Eulerian grid is 1024×1024 , the resolution of the Lagrangian grid ($Im_b \times Jm_b = 251 \times 39$), the penalization parameter $\eta = 5 \times 10^{-4}$ and the kinematic viscosity is equal to $\nu = 1.4 \times 10^{-4}$. Samples of the backbone of the fish are plotted in Fig. 4.25.

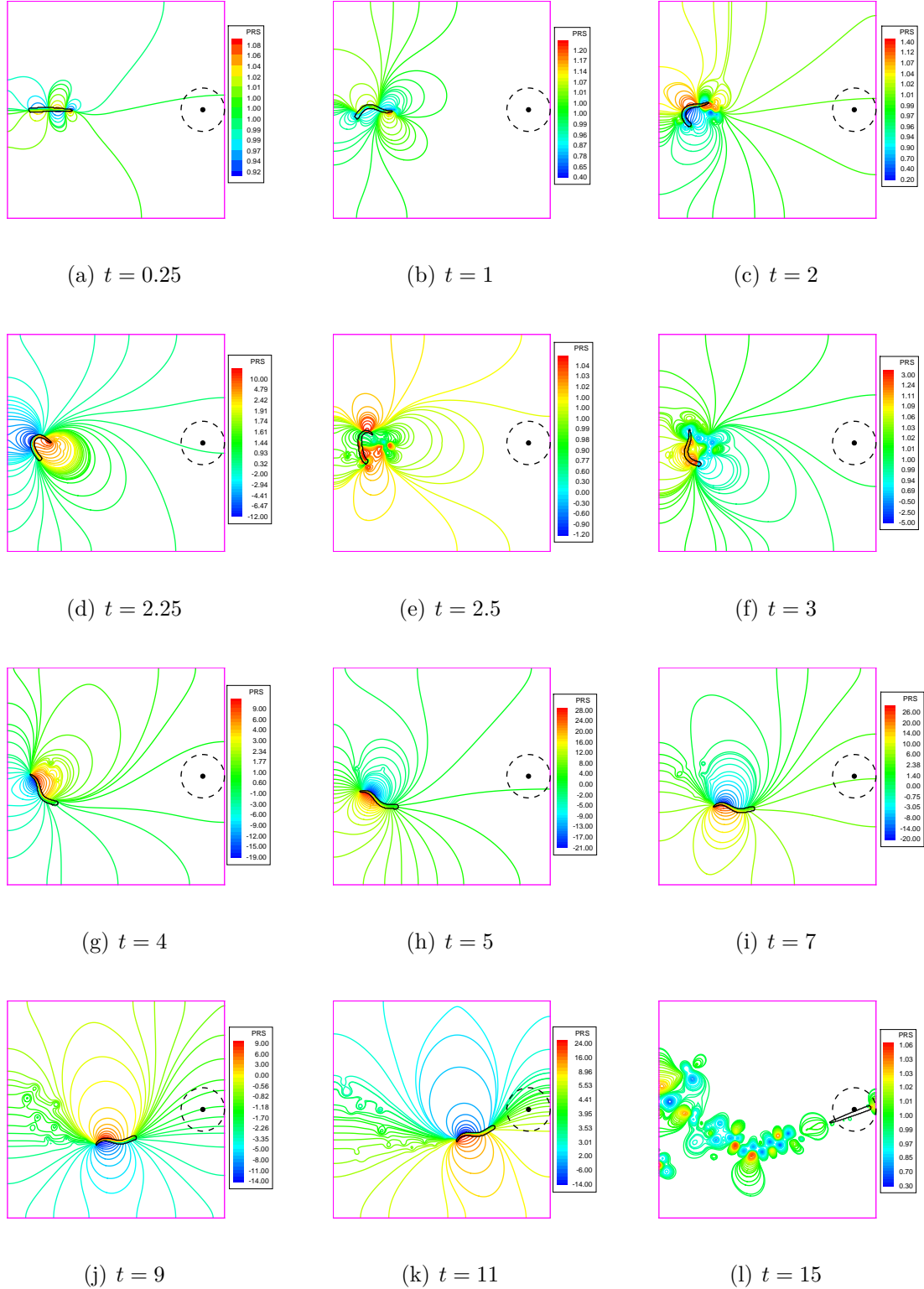


Figure 4.22: Snapshots of pressure isolines obtained during a simulation of swimming fish (represented by black contour corresponding to $\chi = 0.2$) toward a predefined goal which is located at $(x_f, y_f) = (0.9L_x, 0.5L_y)$. At $t = 0$ the fish and the surrounding flow are at rest. After reaching the vicinity ($r = 0.5l_{\text{fish}}$) of the target the curvature of the backbone tends to zero by Eq. (4.26). The domain of the solution is $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, the resolution of the Eulerian grid is 1024×1024 , the resolution of the Lagrangian grid ($Im_b \times Jm_b = 251 \times 39$), the penalization parameter $\eta = 5 \times 10^{-4}$ and the kinematic viscosity is equal to $\nu = 1.4 \times 10^{-4}$.

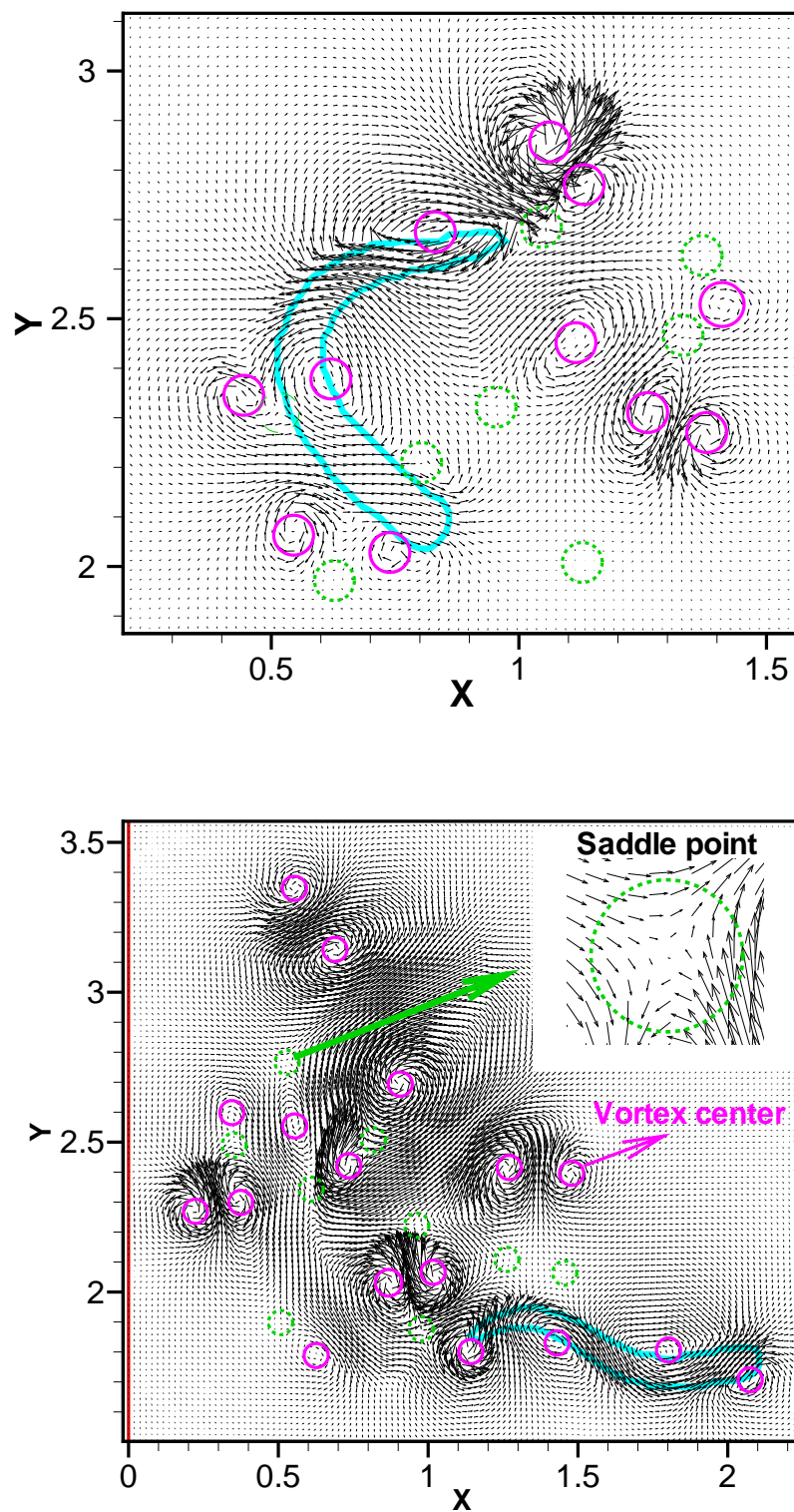


Figure 4.23: Saddle points are denoted by green dashed circles and vortices are denoted by purple solid circles. The vortices forming dipoles during the rotation, corresponding to strong jets.

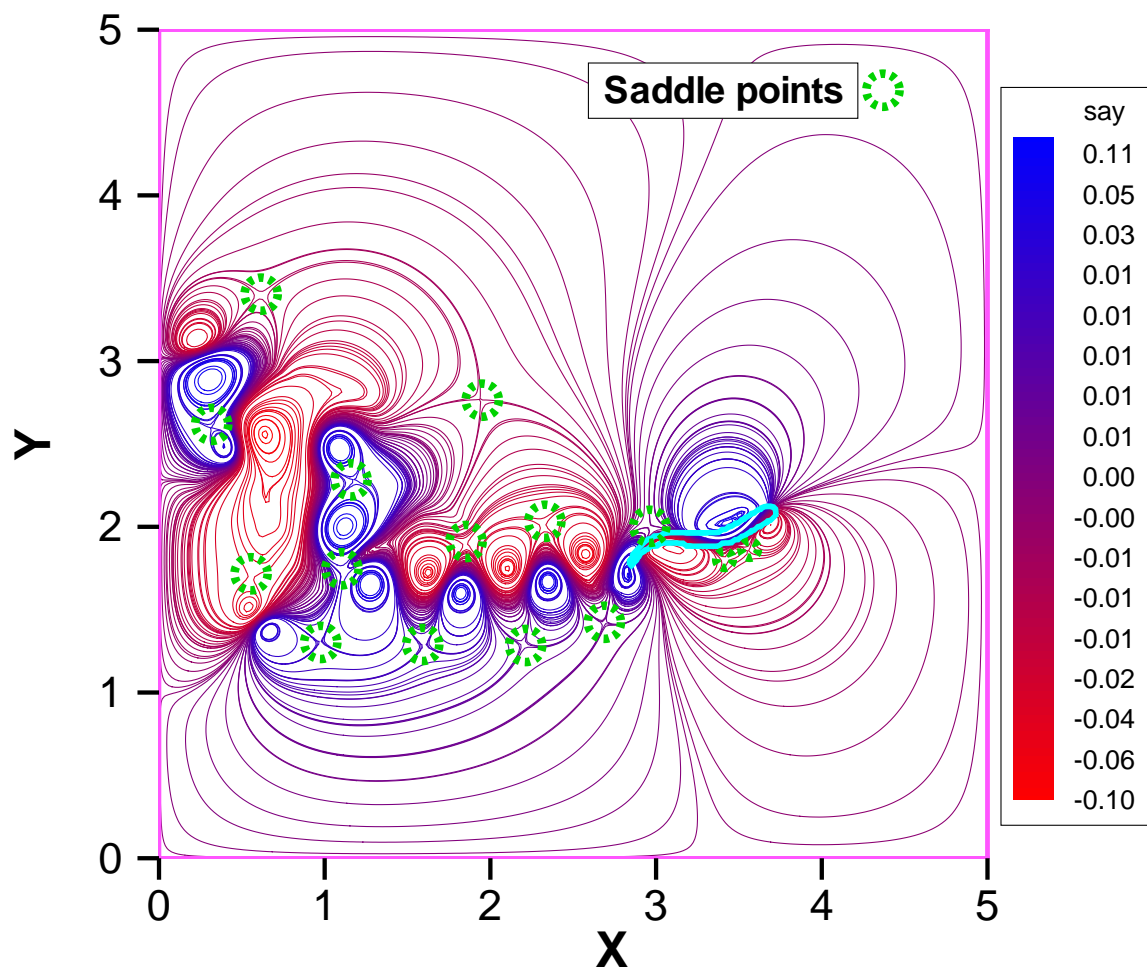


Figure 4.24: Stream-function ψ isolines during the rotation of the fish. Saddle points are denoted by green dashed circles and vortices are denoted by blue (counter-clockwise) and red (clockwise) solid circles.

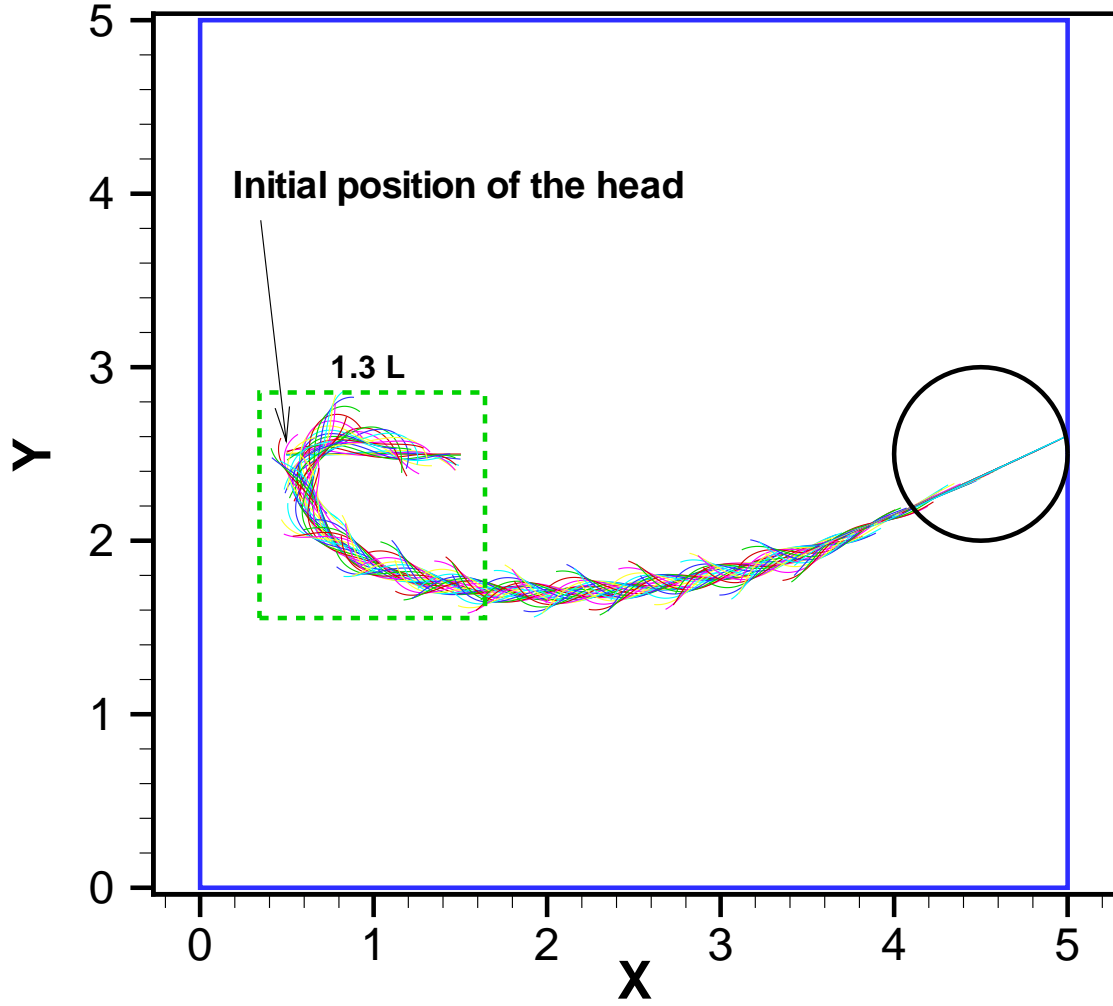


Figure 4.25: Samples of the backbone of a swimming fish toward a predefined goal which is located at $(x_f, y_f) = (0.9L_x, 0.5L_y)$ obtained during a simulation, $t \in [0, 15]$. After reaching the vicinity ($r = 0.5l_{\text{fish}}$) of the goal the curvature of the backbone tends to zero by Eq. (4.26). The snapshots of the corresponding vorticity and pressure isolines are plotted in Figs. 4.21 and 4.22. Starting from rest the fish performs a 180° rotation within an area of about 1.3 times its length.

Chapter 5

Fluid–structure interaction on adaptive grids

“One cannot escape the feeling that these mathematical formulae have an independent existence and an intelligence of their own, that they are wiser even than their discoverers, that we get more out of them than was originally put into them [118].”

Heinrich Hertz

In this Chapter, a space adaptive multiresolution method is developed to solve the incompressible two-dimensional Navier–Stokes equations in vorticity-stream function formulation including the penalization term. The new method is based on a multiresolution analysis which allows to reduce the number of active grid points significantly by refining the grid automatically via nonlinear thresholding of the wavelet coefficients in a one-to-one correspondence with the grid points. To assess the accuracy of the method, the dipole-wall collision, studied by Clercx and Bruneau [104] is considered as a benchmark. A good agreement between the results of the adaptive simulations and those obtained with the uniform grid solver is obtained. The grid adaptation strategy uses an estimation of the local regularity of the solution via wavelet coefficients at a given time step. An extension to interactions with forced deformable bodies, i.e., swimming fish, is done using the volume penalization method. A Lagrangian structured grid with prescribed motion covers the deformable body interacting with the surrounding fluid due to the hydrodynamic forces and the torque calculated on an Eulerian Cartesian reference grid. The results of swimming fish are compared with those of Gazzola et al. [144] where a uniform grid is used. The obtained results show that the CPU-time of the adaptive simulations can be significantly reduced with respect to simulations on a regular grid. Nevertheless the accuracy order of the underlying numerical scheme is preserved.

5.1 General introduction

The aim in this chapter is to develop a reliable self-adaptive numerical method for direct simulation of incompressible flows. Conventional methods for spatial discretization of the PDEs (e.g., finite differences, finite volumes and finite elements) have limited order of accuracy especially near boundaries, but they are more flexible in dealing with complex geometries over a suitable grid. On the other hand standard spectral methods which are widespreadly used in direct numerical simulation of turbulence are limited to Cartesian grids. Spectral elements are a compromise between the two mentioned methods. One can recognize the poor spectral localization (good spatial localization/resolution) of the former methods while the latter have a good spectral localization (poor spatial localization/resolution) [70]. The limitation of the mentioned methods for problems with a wide range of active spatial scales, has encouraged researchers to use alternative methods, e.g., adaptive methods, with limited accuracy but good spatial localization in regions where steep gradients of the flow variables are present. Adaptive methods can be divided into r-type (a fixed number of grid points are redistributed), h-type (regridding is performed occasionally) and p-type (the degree of the polynomial representing the solution is locally increased), each one with their own advantages and disadvantages as detailed in the literature. Among different methods for grid adaptation h-type refinement proved to be more advantageous in terms of error control. Between different error-estimating adaptation strategies (which most of them belong to the finite element family) wavelet-based numerical methods have proved to be an efficient tool in developing adaptive numerical methods which control the global (usually L_2) approximation error. Wavelet transforms allow to estimate the local regularity of the solution to a given PDE, using an efficient algorithm, and thus can define auto-adaptive discretizations with local mesh refinement [107]. Lian-drat and Tchamitchian [49] proposed the first wavelet-based adaptive method for numerical simulation of PDEs. The currently existing wavelet-based algorithms can be classified as pure wavelet methods and wavelet optimized grid methods. Pure wavelet methods, employ wavelets directly for discretization of the governing equations. On the other hand, wavelet optimized grid (WOG) methods [109] combine classical discretizations of considered equations (e.g., finite differences or finite volumes) with wavelets, which are used to define the adaptive grid. We refer to [89] and [136] where a finite volume discretization of the governing equations combined with a cell-averaged interpolating wavelet transform for grid adaptation is used. For more details we refer to the review paper by Schneider and Vasilyev [141]. In the present work the method of adaptive multiresolution analysis is applied to the Navier–Stokes equations in the vorticity and stream-function formulation. However, the concepts are also applicable to the primitive variable formulation. Thus similar to WOG methods the role of the wavelet transform is the adaptation of the grid and the fast interpolation of the flow variables at newly inserted points, necessary to account for

the flow evolution. A second-order central finite difference method with symmetric stencil over an adaptive Cartesian grid is used for spatial discretization of the equations. The finite difference method represents a suitable combination with the multiresolution analysis based on Harten's point-value wavelet transform. The concept of symmetric stencils leads to intermediate (hung) points, that their values can be interpolated accurately via the inverse wavelet transform, for details we refer to Durbin and Iaccarino [88]. After validation of the developed adaptive multiresolution solver, using the results of previous studies of dipole-wall collision, an extension to fish swimming with the volume penalization method is presented. The volume penalization method is a sub branch of immersed boundary methods, see [99] for a complete review of these methods. As starting point in the present work we take the two-dimensional vorticity stream-function solver developed by Sabetghadam et al. [127] for a uniform grid and the adaptive solver developed in [143] for simulation of the flow inside curved geometries. The code is developed in FORTRAN and is open access [172]. This chapter is organized as follows; First a summary of the governing equations, the multiresolution analysis, the discrete wavelet transforms and the idea of point selection by filtering of the wavelet coefficients will be presented. After that, for validation of the solver the results of the dipole-wall collision are compared with previous studies. Next a test case for swimming fish will be demonstrated as application for fluid-structure interaction on adaptive grids. Finally, the results and some perspectives will be discussed.

5.2 Multiresolution analysis of incompressible flow

The governing equations of incompressible flows in two-dimensions can be reformulated in the vorticity ω and stream-function ψ form (see Section 2.1). Denoting by $E(\Delta t)$ the discrete time evolution operator, the global algorithm can be schematically summarized by

$$\omega^{n+1} = E(\Delta t) \left[M^{-1} \cdot S \cdot T(\epsilon) \cdot M \right] \omega^n \quad (5.1)$$

where M and M^{-1} are the direct (WT) and inverse (IWT) wavelet transform operators. $T(\epsilon)$ is the thresholding operator and S represents the safety zone operator. For an explicit Euler time integration we have

$$E(\Delta t)\omega^n = \omega^n + \Delta t \, RHS(\omega^n). \quad (5.2)$$

where the RHS operator contains all the terms (spatial derivatives) of the vorticity equation (2.7) except the time derivative. The summary of the multiresolution method is given in Algorithm 3. Some necessary criteria that must be respected in the algorithm, are given in the following:

1. Before interpolation of the values of the independent variables via IWT (from the coarsest level up to the finest level) in some grid points (with setting wavelet coefficients equal to zero, $d = 0$), it is necessary to mark all the intermediate necessary points for having a consistent WT, (from the finest level down to the coarsest level) and adding them to the list of the points to be interpolated.
2. In the time integration step, by using multi-step schemes, e.g., the Runge–Kutta family, before calculation of spatial derivatives at intermediate steps, the value of u^* at the hung points must be interpolated again from the new values of the active points. Moreover 6-(a) and 6-(b) will be done once in each time step.
3. In the case of the two-dimensional Navier–Stokes equations in vorticity-stream function formulation, before calculation of the spatial derivatives it is necessary to solve an elliptic equation, i.e., Eq. (2.6) for updating the stream-function. For more details see [169].

5.2.1 Biorthogonal wavelet transform

To explain the concept of WT, we consider the case of Harten’s point values representation [64] on uniform grids, which is well adapted for finite difference methods, in contrast to Harten’s cell average method which is suitable for finite volume methods. By considering a unit interval $[0, 1]$, the hierarchy of uniform dyadic grids is obtained from

$$X_j = \{x_{j,i} \in \mathbb{R} : x_{j,i} = i2^{-j}, i = 0, \dots, 2^j\}, j = 0, \dots, J \quad (5.3)$$

with spacing 2^{-j} , where j is the level and i represents the position index. The number of points must always be odd ($N = 2^J + 1$) to have a point in the middle. A given discrete function $f(x)$ can be represented with the use of a wavelet basis as follows

$$f(x) = \sum_{i=0}^{2^J} f_{0,i} \Phi_{0,i}(x) + \sum_{j=0}^J \sum_{i=0}^{2^j} d_{j,i} \Psi_{j,i}(x) \quad (5.4)$$

where the bases consist of scaling functions $\Phi_{j,i}$ and wavelets $\Psi_{j,i}$. Interpolating wavelet coefficients are defined as

$$d_{j,i} = \langle f, \Psi_{j,i} \rangle = f_{j+1,2i+1} - \tilde{f}_{j+1,2i+1} \quad (5.5)$$

where cubic (third-order) interpolation can be used as follows,

$$\tilde{f}_{j+1,2i+1} = \frac{-f_{j,i-1} + 9f_{j,i} + 9f_{j,i+1} - f_{j,i+2}}{16} \quad (5.6)$$

Algorithm 3 Multiresolution analysis

1. Start from an initial condition given on a dyadic grid
 2. Apply WT to the active points (from the finest level down to the coarsest level), to compute the wavelet coefficients of the independent variable
 3. Perform thresholding $T(\epsilon)$, to remove all points from the list of active points having a magnitude of wavelet coefficients below the corresponding threshold ϵ_j
 4. Add safety zone to the list of new active points
 - (a) Add neighbor points at the same level and one level above
 - (b) Guarantee the gradedness of the new active points (optional)
 - (c) Add necessary points to the current list of active points, for having a consistent direct or inverse WT
 5. Apply IWT to the new active points to compute the values of the independent variables (or interpolate the values of all newly added points via IWT with zero wavelet coefficient $d = 0$)
 6. Perform the time evolution of the independent variable for all the active points
 - (a) Search for the nearest active point to determine $dist$ for all active points
 - (b) Check for the existence of all other neighbors of the active points with distance $dist$, mark all the missing points as hung points
 - (c) Interpolate the values of the hung points via IWT with zero wavelet coefficient $d = 0$
 - (d) Compute the spatial derivatives for the given PDE via FDM with symmetric stencils
 7. Go to step 2, if $T < T_{end}$
-

Near boundaries forward/backward stencils must be used for interpolation. For the two-dimensional transform and more details we refer to Ghaffari and Schneider [169].

5.2.2 Filtering of wavelet coefficients

Given a threshold parameter for the finest ϵ_J or the coarsest level ϵ_0 , data compression can be obtained, by thresholding of the detail coefficients. This procedure also called nonlinear filtering of the wavelet coefficients. After performing the direct transform, wavelet coefficients with magnitude smaller than the threshold are set to zero and the corresponding points can be eliminated from the set of the points. In other words we can find the value of that point by interpolation and the error remains bounded by the threshold value. The thresholding defined as

$$d_{j,i} = \begin{cases} 0 & \text{if } |d_{j,i}| \leq \epsilon_j, \\ d_{j,i} & \text{else} \end{cases} \quad (5.7)$$

where $\epsilon_j = \epsilon_J 2^{D(j-J)} = \epsilon_0 2^{D(j)}$, $D = 1, 2, 3$ is the dimension of the problem, and J denotes the maximum level. After nonlinear filtering in wavelet space the given function $f(x)$, can be reconstructed $\bar{f}(x)$, just with the significant wavelet coefficients corresponding to the points where the function is less regular. Those points must be kept to guaranty the boundedness of the error introduced due to filtering and eliminating non necessary points. Following Donoho [54], it can be shown that for a sufficiently smooth function $f(x)$, the error is bounded by the threshold, i.e., $|f(x) - \bar{f}(x)| \leq c_1 \epsilon_0$, where c_i is a constant. For illustration of the idea we consider a non-periodic one-dimensional function $f(x) \in [0, 1]$ defined as follows

$$f(x) = \begin{cases} 8.1e^{1/4}e^{-|x-1/2|} & 0.0 \leq x < 0.25 \\ 9e^{-|x-1/2|} & 0.25 \leq x < 0.75 \\ e^{-|x-1/2|}(16x^2 - 24x + 18) & 0.75 \leq x \leq 1.0 \end{cases} \quad (5.8)$$

with a jump in value located at $x = 0.25$, a jump in the value of the first derivative at $x = 0.5$ and a jump in the value of the second derivative at $x = 0.75$. We consider also a Gaussian function, $f(x) = \exp(-(x - 0.5)^2/\delta^2)$ where $x \in [0, 1]$. Their sparse point representations, with the use of a cubic interpolating wavelet transform ($P_{WT} = 4$), for $J = 10$, filtered with a threshold $\epsilon = 1 \times 10^{-3}$ are illustrated in Fig. 5.1 (a) and (b). A good compression and an error bounded by the threshold can be seen, for more details see [169].

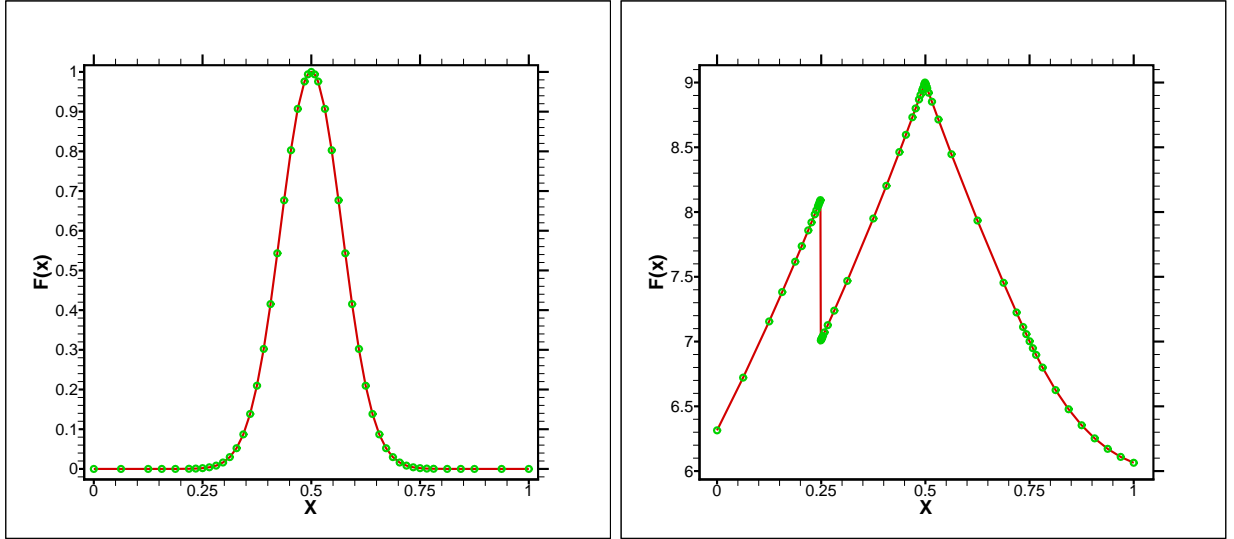


Figure 5.1: Sparse point representation of 1D functions, obtained by WT with cubic interpolation ($J = 10$), filtered with threshold $\epsilon = 1 \times 10^{-3}$. The green dots (marked ●) show the retained grid points. Left: Gaussian function, compression = 95%, L_∞ -Error $\leq 1 \times 10^{-4}$. Right: Function (5.8), Compression = 94%, L_∞ -Error $\leq 5 \times 10^{-5}$.

5.3 Dipole-wall collision with a uniform grid solver

In this section we will study the classical benchmark studied by Clercx and Bruneau [104] and Keetels et al. [112]. The flow is confined in a square domain $(x, y) \in [0, 2] \times [0, 2]$. At the four walls of the domain ($x, y = 0$ & $x, y = 2$) no-slip and no-penetration boundary conditions are applied. The flow is initialized in the form of two shielded Gaussian monopolar vortices, which their centers placed at a distance 0.2 apart. The vorticity distribution of each monopole is given by

$$\omega(0, \mathbf{x}_n) = \omega_e \left(1 - \frac{r^2}{r_0^2} \right) \exp \left(- \frac{r^2}{r_0^2} \right) \quad (5.9)$$

where r_0 is the core radius, $r = \|\mathbf{x} - \mathbf{x}_n\|$ with \mathbf{x}_n being the position of the vortex center. The two isolated monopoles are located at

$$\mathbf{x}_1 = (1, 1.1) \text{ and } \mathbf{x}_2 = (1, 0.9)$$

The initial vorticity field is the sum of two vorticity fields given by Eq. (5.9). The core radius of the shielded monopoles are set to $r_0 = 0.1$. Following Keetels et al. [112], demanding that the root mean square (rms) velocity is initially equal to unity, yields the amplitude of the isolated monopole $\omega_e = \pm 299.528385375226$. The total normalized initial energy by considering the surface of the domain is equal to $E = 2$. The vorticity amplitude in the radial direction decreases exponentially with r . As a result, the circulation of one

isolated monopole calculated over a circular contour around the vortex origin decreases exponentially towards zero for increasing contour radius. Hence, no boundary layers are required at the no-slip walls when constructing the initial flow field. The integral-scale Reynolds number for the initial field is given by $Re = U_{rms}L/\nu$, where the characteristic length scale is set to the half-height of the domain, here $L = 1$, and the characteristic velocity to the initial root mean square velocity, $U_{rms} = 1$. This integral Reynolds number differs slightly from the Reynolds number $Re_d \approx 0.8Re$ based on the dipole translation speed U_d and the dipole radius R , we refer to Kramer [113] for more details. To verify the accuracy of the numerical method the results are compared with those reported by Clercx and Bruneau [104] that are computed with a pseudo-spectral solver. In Clercx and Bruneau [104] both the velocity and vorticity are expanded in a truncated series of Fourier polynomials for the periodic-direction and in a truncated series of Chebyshev polynomials for the non periodic-direction. In the case of dipole-wall collision the boundary conditions are not periodic, thus Chebyshev polynomials must be used. Some invariants of the flow in periodic or unbounded domains, i.e. total energy and enstrophy, which are conserved by the flow dynamics for inviscid fluids ($\nu = 0$) can be assessed in viscous flows, where they will not be conserved, but instead varying in time, depending on the Reynolds number. Three integral quantities in the flow field, i.e., total energy E , total enstrophy Z and total palinstrophy are defined as:

$$E(t) = \frac{1}{2} \int_{\Omega} |\mathbf{u}(\mathbf{x}, t)|^2 d\mathbf{x} \approx \frac{\Delta x \Delta y}{2} \sum_{i=1}^{Imax} \sum_{j=1}^{Jmax} (u_{i,j}^2 + v_{i,j}^2) \quad (5.10)$$

$$Z(t) = \frac{1}{2} \int_{\Omega} |\omega(\mathbf{x}, t)|^2 d\mathbf{x} \approx \frac{\Delta x \Delta y}{2} \sum_{i=1}^{Imax} \sum_{j=1}^{Jmax} (\omega_{i,j})^2 \quad (5.11)$$

$$P(t) = \frac{1}{2} \int_{\Omega} |\nabla \omega(\mathbf{x}, t)|^2 d\mathbf{x} \approx \frac{\Delta x \Delta y}{2} \sum_{i=1}^{Imax} \sum_{j=1}^{Jmax} \left(\frac{\partial \omega_{i,j}}{\partial x} \right)^2 + \left(\frac{\partial \omega_{i,j}}{\partial y} \right)^2 \quad (5.12)$$

The approximations are obtained by applying the trapezoidal quadrature formula for two-dimensions. Following Kraichnan and Montgomery [38] for any two-dimensional viscous flow ($Z > 0$) the total energy $E(t)$ decays according to

$$\frac{dE}{dt} = -\nu \int_{\Omega} \omega^2 dA = -2\nu Z. \quad (5.13)$$

where $-2\nu Z$ is the energy dissipation. Note that the decay rate is proportional to the total enstrophy $Z(t)$, which is a measure of the squared vorticity integrated over the domain. Understanding the evolution of the total enstrophy is therefore of crucial importance for explaining the energy decay. For a domain with no-slip boundaries the change in total

enstrophy is governed by

$$\frac{dZ}{dt} = -2\nu P + \nu \oint_{\partial\Omega} \omega(\mathbf{n} \cdot \nabla \omega) ds \quad (5.14)$$

where \mathbf{n} denotes the outward unit normal vector with respect to $\partial\Omega$. The first term on the right-hand side simply states that the enstrophy decays due to vorticity gradients (palinstrophy) that are present in the flow. The second term represents the vorticity production at the no-slip boundaries involving the vorticity and its gradients, which will give rise to the total palinstrophy. Note that the vorticity influx at the no-slip boundaries is equal to $(\mathbf{n} \cdot \nabla \omega)$. In the case of a square domain with stress-free or periodic boundary conditions the second term on the right-hand side of Eq.(5.14) vanishes. As a result, the total enstrophy cannot increase for a domain with stress-free or periodic boundary conditions and is thus always bounded by its initial value and zero [113]. We refer to the discussion by Clercx and Heijst [87] and also by Schneider and Farge [102]. For a steady flow we have:

$$P = \frac{1}{2} \oint_{\partial\Omega} \omega(\mathbf{n} \cdot \nabla \omega) ds$$

Different simulations by successively increasing the number of points (256^2 , 512^2 and 1024^2) were performed to obtain a grid independent solution. The simulations were performed on a uniform grid with a second order finite difference multi-grid solver. To have a stable simulation the time step must be reduced according to the CFL condition. The evolution of the total kinetic energy, total enstrophy and total palinstrophy for $Re = 1000$ are compared with the computations of Clercx and Bruneau [104] in Fig. 5.2 (b), (c) and (d), respectively. Note that the energy steadily decreases from its normalized initial value of $E = 2$ towards $E \approx 0.8$ at $t = 1$. At $t \approx 0.35$ the kinetic energy decays faster, which is due to the increased enstrophy production (dissipation) at the boundaries of the domain. The first peak in the enstrophy curve takes place at $t = 0.35$, and thus coincides with the first collision of the dipole with the right wall, see Fig. 5.2 (a). During this first collision the boundary layers create a large amount of vorticity. The enstrophy in the boundary layers is then the main contribution to the total enstrophy. At $t \approx 0.64$ another smaller peak is visible in the enstrophy evolution curve, which is due to the second collision of newly generated vortices with the right wall. The results for the total energy are in reasonable agreement with the pseudo-spectral simulations, a maximum difference in energy less than 4% can be seen. We can see a systematic deviation in enstrophy and palinstrophy curves especially near the first peak, which is decreasing by increasing the resolution in comparison with the reference solution of Clercx and Bruneau [104]. See also the discussion by Nguyen et al. [151].

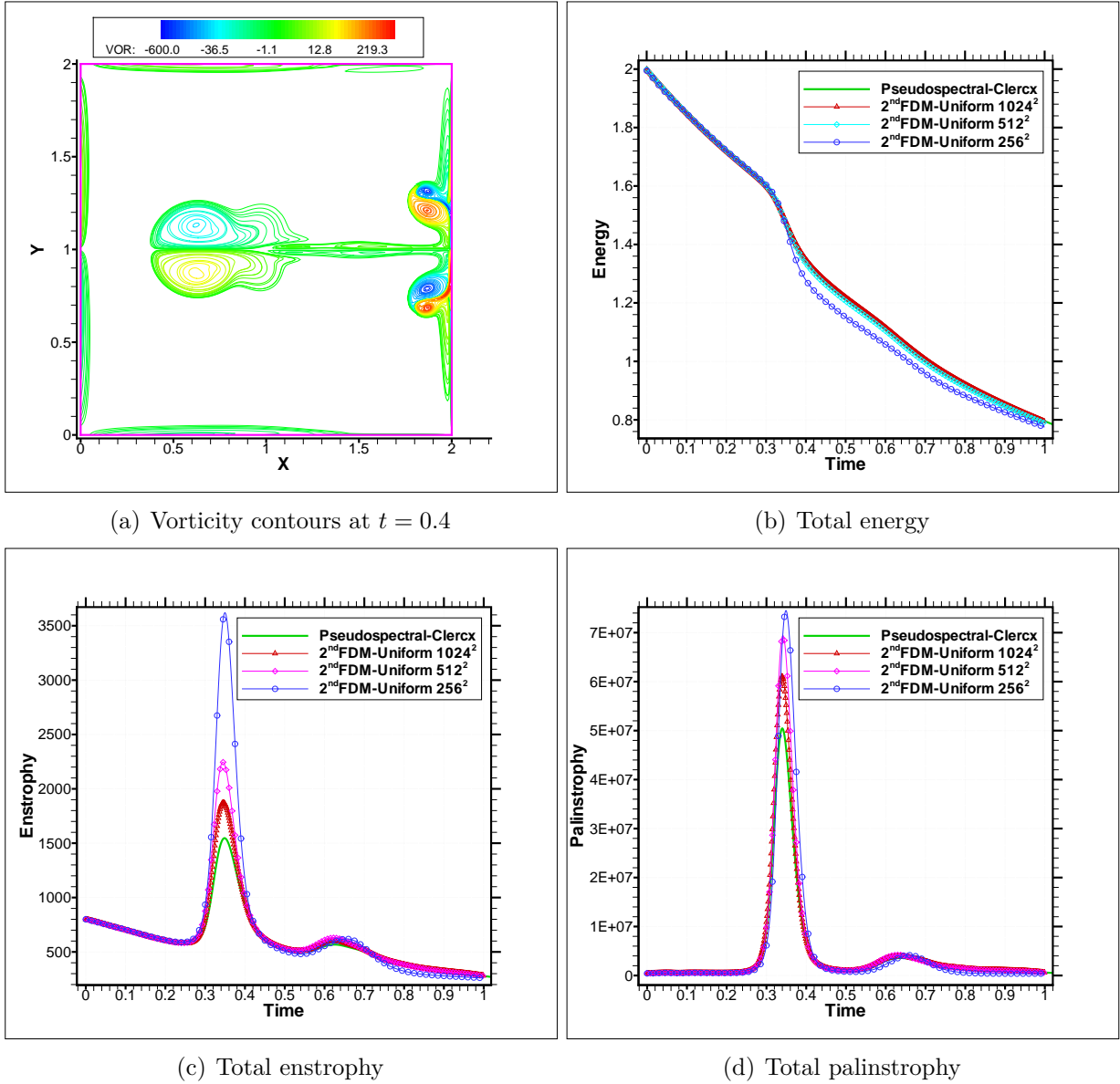


Figure 5.2: (a) Vorticity contours of dipole-wall collision at $t = 0.4$. (b)-(d) Comparison of the total energy $E(t)$, the total enstrophy $Z(t)$ and the total palinstrophy $P(t)$ between the data from Clercx and Bruneau [104] and the present finite-difference computation with a uniform multigrid solver for Reynolds 1000 with different grid resolutions. By increasing the resolution the results converge toward the reference simulation.

5.4 Validation of the adaptive multiresolution solver

In this section the problem of dipole-wall collision studied by Clercx and Bruneau [104] is chosen as a benchmark computation for validation of the proposed algorithm. The time evolution of the dipole is calculated by the developed multiresolution finite difference solver with threshold $\epsilon = 10^{-3}$ and a maximum grid level $J = 11$ for Reynolds 1000. The evolution of the vorticity isolines and the corresponding adaptive grid starting from the initial condition at $t = 0$ up to $t = 1$, are shown in Fig. 5.3. Comparisons of the total energy $E(t)$ and the total enstrophy $Z(t)$ between the uniform grid solver and the multiresolution computation with thresholds, $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$, with maximum grid level $J = 9$ are plotted in Fig. 5.4 (a) and (b), respectively. The agreement between the uniform grid solver and the multiresolution solver is good and the results for $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$ are almost identical. Therefore we will use $\epsilon = 10^{-3}$ for all multiresolution computations. A convergence study for the total enstrophy $Z(t)$ (with the uniform grid solver) for Reynolds 1000, with different grid spacings, i.e., a maximum level in each direction $J = 8, 9, 10, 11$, is performed. Once again the simulation with the pseudo-spectral solver of Clercx and Bruneau is taken as reference solution [104]. The results of the present computations are illustrated in Fig. 5.4 (c). It can be observed that by increasing the number of grid points the curves get closer and closer. We anticipate that the results of $J = 12$ will match those of Clercx and Bruneau [104] for enstrophy.

5.5 Application to fish-like swimming

Anguilliform swimming presented in Gazzola et al. [144] is considered as application for the proposed algorithm. Details of our fluid/solid interaction algorithm are given in [170]. A periodic swimming law is defined by fitting the backbone of the fish to a given curve $y(x, t)$ keeping the backbone length l_{fish} fixed. Let ξ be the arclength of the curvilinear coordinate of the deformed backbone ($0 \leq \xi \leq l_{\text{fish}}$). For uniformly distributed points $\Delta\xi = l_{\text{fish}}/(N - 1)$ over the backbone, y is given by

$$y(x, t) = a(x) \sin(2\pi(x/\lambda + ft)) \quad (5.15)$$

where λ is the wavelength, f is the frequency of the backbone and $a(x)$ is the envelope $a(x) = a_0 + a_1x + a_2x^2$ where x is defined by inverting the arclength, i.e., $\Delta x = \Delta\xi/\sqrt{1 + (\partial y/\partial x)^2}$. The geometry of the fish is given by Eq. (4.6). The parameters used by Gazzola et al. [144] for the kinematics of the fish are as follows: $\lambda = 1$, $f = 1$, $a_2 = 0$, $a_1 = 0.125/(1 + c)$, $a_0 = 0.125c/(1 + c)$ and $c = 0.03125$. The buoyancy is equal to zero, i.e., $\rho_b = \rho_f$. The viscosity of the fluid is set to $\nu = 1.4 \times 10^{-4}$ resulting in a Reynolds number of approximately $Re \approx 3800$, based on asymptotic mean velocity of

$U_{\text{forward}} \approx 0.52$. The simulations of Gazzola et al. [144] are carried out on a rectangular domain $(x, y) \in [0, 8l_{\text{fish}}] \times [0, 4l_{\text{fish}}]$ with a resolution of 4096×2048 grid points and a penalization parameter of $\eta = 10^{-4}$. We are performing our simulations in a square domain $(x, y) \in [0, 8l_{\text{fish}}] \times [0, 8l_{\text{fish}}]$ by imposing a penalization parameter inside the body equal to $\eta = 10^{-3}$ with maximum resolution of 1025×1025 grid points and $\Delta t = 5 \times 10^{-4}$. The centroid of the fish is initially positioned at $x_{\text{cg}} = 0.9L_x$ and $y_{\text{cg}} = 0.5L_y$ in our simulations. We impose two degrees of freedom fixing the angular velocity equal to zero. The simulations start with the body and fluid at rest. The forward velocities of the center of the mass computed with different methods/parameters are compared in Fig. 5.5 (left). The evolution of the number of active, significant (corresponding to the retained points after thresholding the wavelet coefficients), safety zone, hung and interpolated points for the wavelet transform during the computation with the multiresolution solver is illustrated in Fig. 5.5 (right). The number of points used in the multiresolution analysis on the uniform grid results in a compression of more than 95%. Fig. 5.6 gives different views of the adaptive grids colored by vorticity and the mask function χ at $t = 6$. The snapshots of the adaptive grid obtained with the multiresolution solver during the simulations from $t = 1$ to 6 are illustrated in Fig. 5.7.

5.6 Conclusion

In the present investigation, a space adaptive multiresolution method was developed to deal with two-dimensional unsteady incompressible flows. The new adaptive method is based on a multiresolution analysis which allows to reduce the number of active grid points significantly by refining the grid automatically via nonlinear thresholding of the wavelet coefficients in a one-to-one correspondence with the grid points. In the present work the concept of adaptive multiresolution technique is applied to the vorticity stream-function formulation of the Navier–Stokes equations. A second-order central finite difference method with symmetric stencil on an adaptive Cartesian grid is used for spatial discretization of the equations. After validation of the proposed algorithm an extension to deal with fluid interaction with forced deformable bodies, i.e., a swimming fish, is done using the volume penalization method. A Lagrangian structured grid with prescribed motion covers the deformable body interacting with the surrounding fluid due to hydrodynamic forces and the torque calculated on an Eulerian reference Cartesian grid. The results of swimming fish are compared with those of Gazzola et al. where a uniform grid is used. The obtained results show that the CPU-time of the adaptive simulations can indeed be reduced with respect to simulations on a uniform grid. The CPU-time reduction depends strongly to the flow configuration, in general the algorithm performs better for external flows. Nevertheless the accuracy order of the underlying numerical scheme is preserved. We state that the multiresolution solver is adaptive in the sense that the CPU-time is reduced by excluding

the unnecessary grid points from the computations by filtering of the wavelet coefficients. In the present investigation there is no memory compression as a classical uniform data structure is used. Implementation of a data-structure for memory deallocation is highly recommended in future developments. In this regard a tree data structure, a hash table or Hilbert type space-filling curves can be used. We refer to Roussel [89], Brix et al. [130] and the discussion by Hejazialhosseini et al. [136]. The code is developed in FORTRAN and is accessible for all [172].

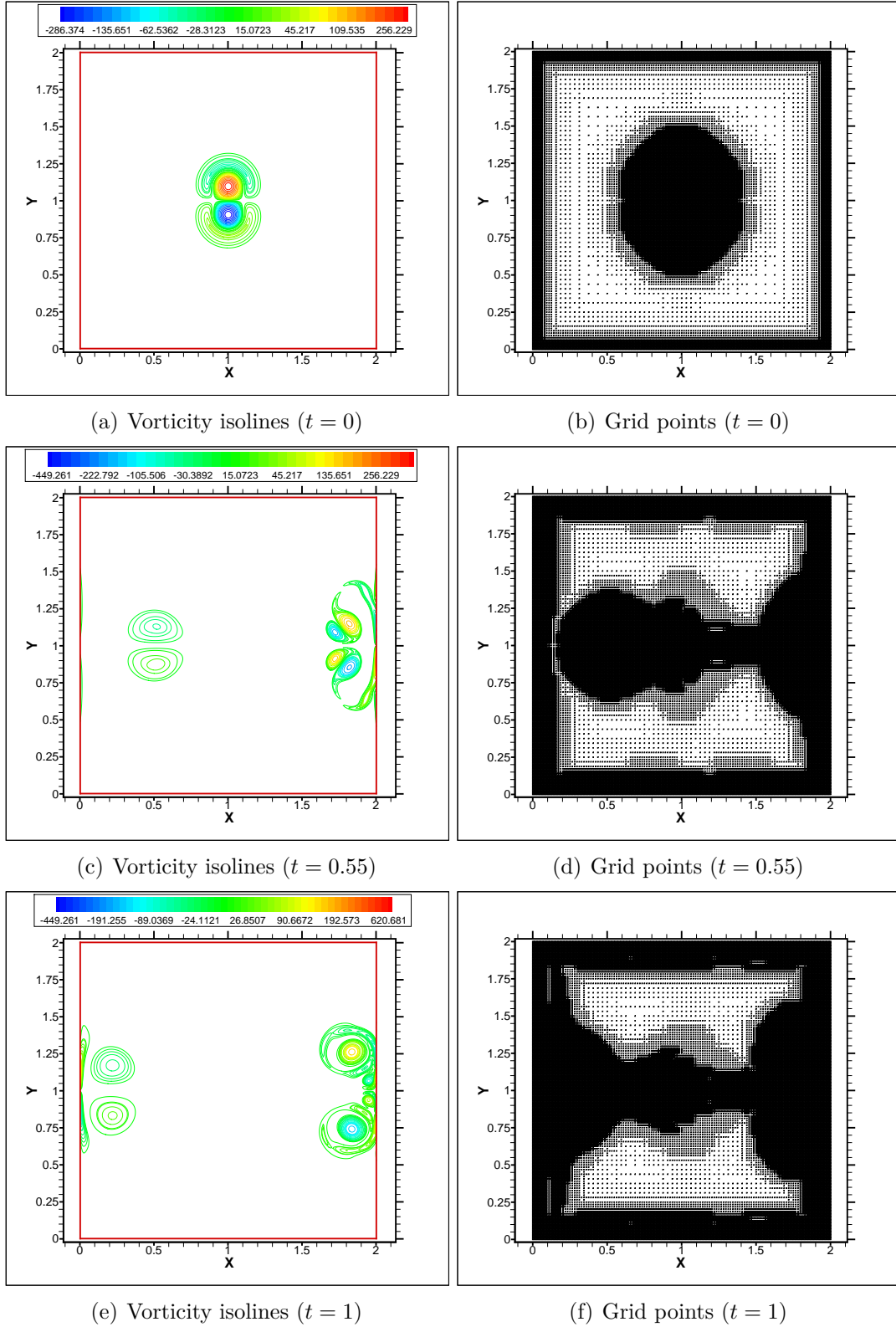


Figure 5.3: The evolution and collision of the vortices (represented by colored isolines) with walls (left) and the corresponding adaptive grid (right), the maximum grid level is $J = 11$ in each direction, threshold $\epsilon = 10^{-3}$, and Reynolds 1000.

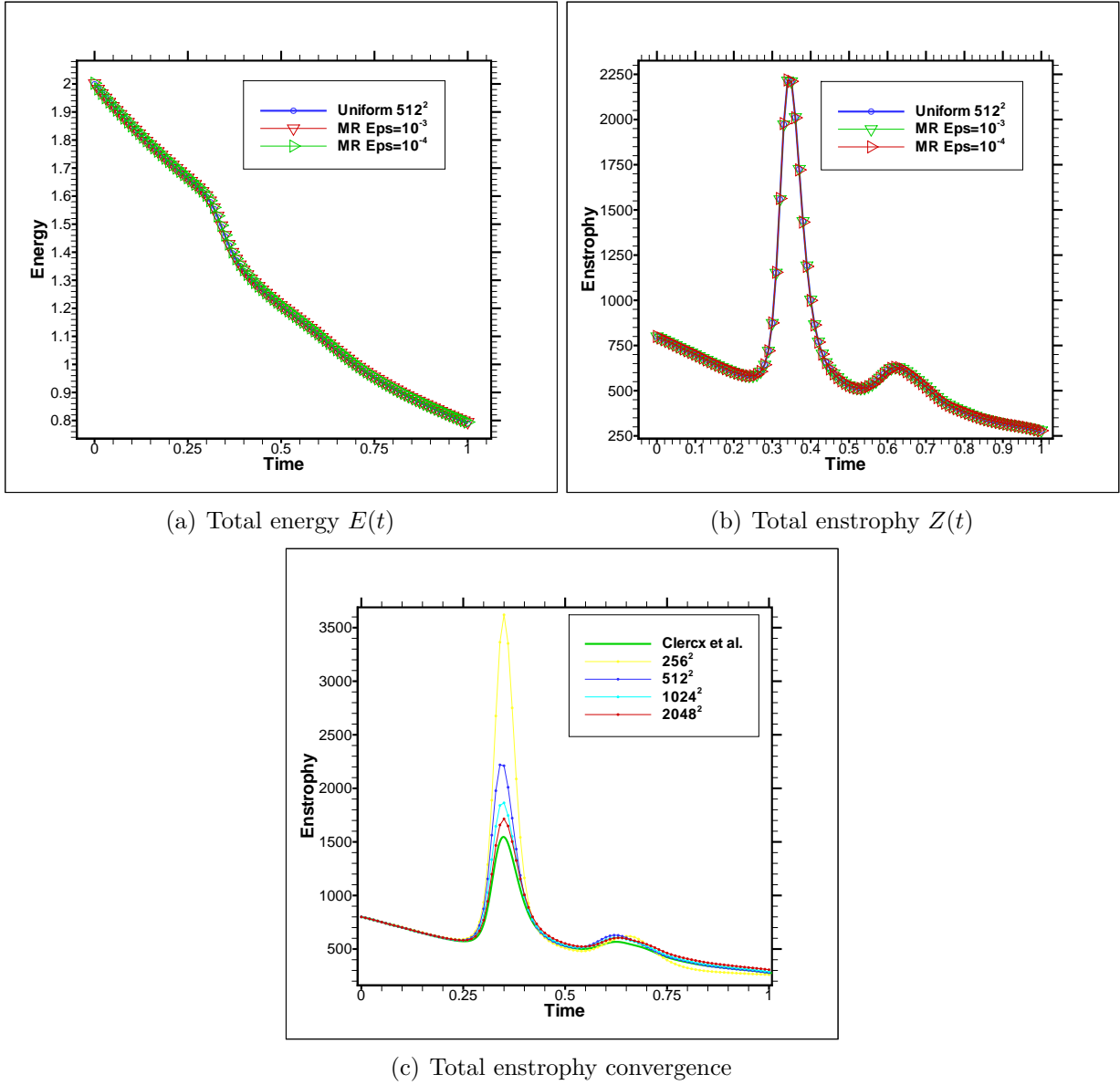


Figure 5.4: Comparisons of the total energy (a) and the total enstrophy (b) between the uniform grid solver and the multiresolution computation with thresholds, $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$, for Reynolds number 1000 and a maximum grid level $J = 9$ in each direction for all simulations. (c) Convergence study for the total enstrophy $Z(t)$ toward the data from Clercx and Bruneau [104] with the uniform grid solver for 256^2 , 512^2 and 1024^2 grid points and multiresolution computations with a maximum grid level $J = 11$ in each direction, for Reynolds 1000. It can be observed that coarse grid computations are too dissipative, we anticipate that 4098^2 grid resolution is sufficient to reproduce the results of Clercx and Bruneau [104].

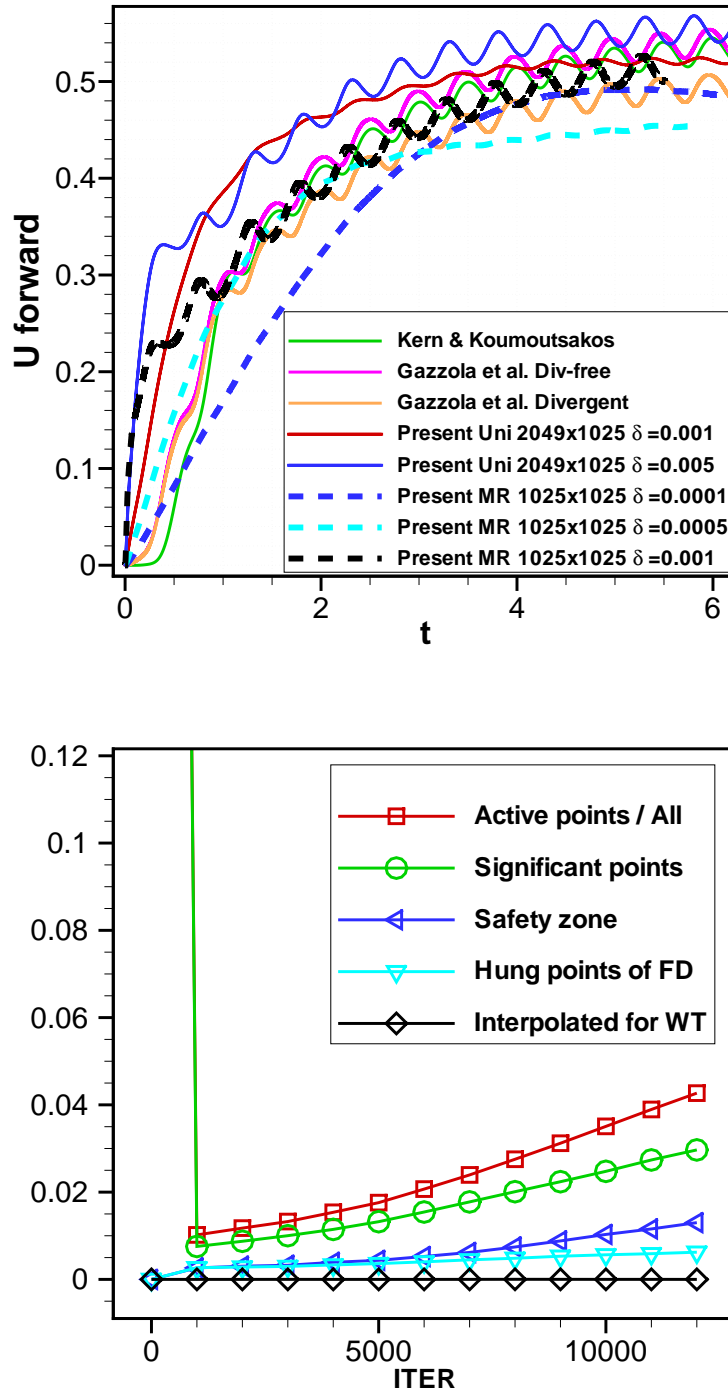


Figure 5.5: Up: Forward velocity U of the anguilliform 2D swimmer ($\lambda = f = 1$). Solid lines indicate the reference simulations performed by Kern and Koumoutsakos (green) [105], Gazzola et al. (pink and brown) [144] and Ghaffari et al. (red and blue) [170]. Dashed lines represent the results with the proposed multiresolution algorithm. Down: Evolution of the number of active grid points, significant points (corresponding to the retained points after filtering of wavelet coefficients), safety zone, hung and interpolated points for the wavelet transform during the computation with the multiresolution solver, with a maximum grid level $J = 10$ in each direction (1025^2 grid points).

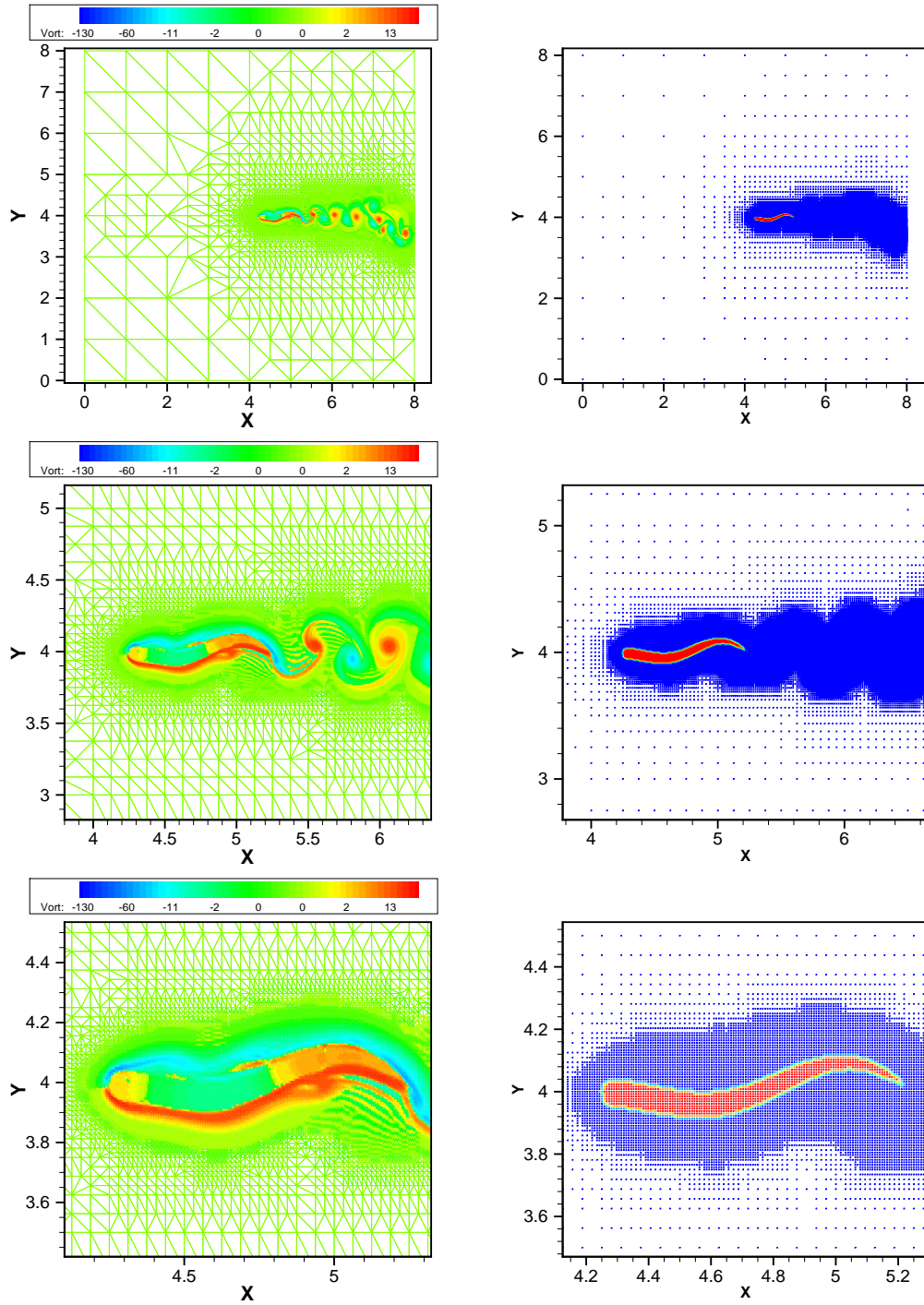


Figure 5.6: Adaptive grids colored by the vorticity (left) and colored by the mask $\bar{\chi}$ (right) at $t = 6$ (zoom in, from up to down) where $(x, y) \in [0, 8l_{\text{fish}}] \times [0, 8l_{\text{fish}}]$ by imposing the penalization parameter inside the body equal to $\eta = 10^{-3}$, with maximum grid level of $J = 10$ in each direction and the viscosity $\nu = 1.4 \times 10^{-4}$.

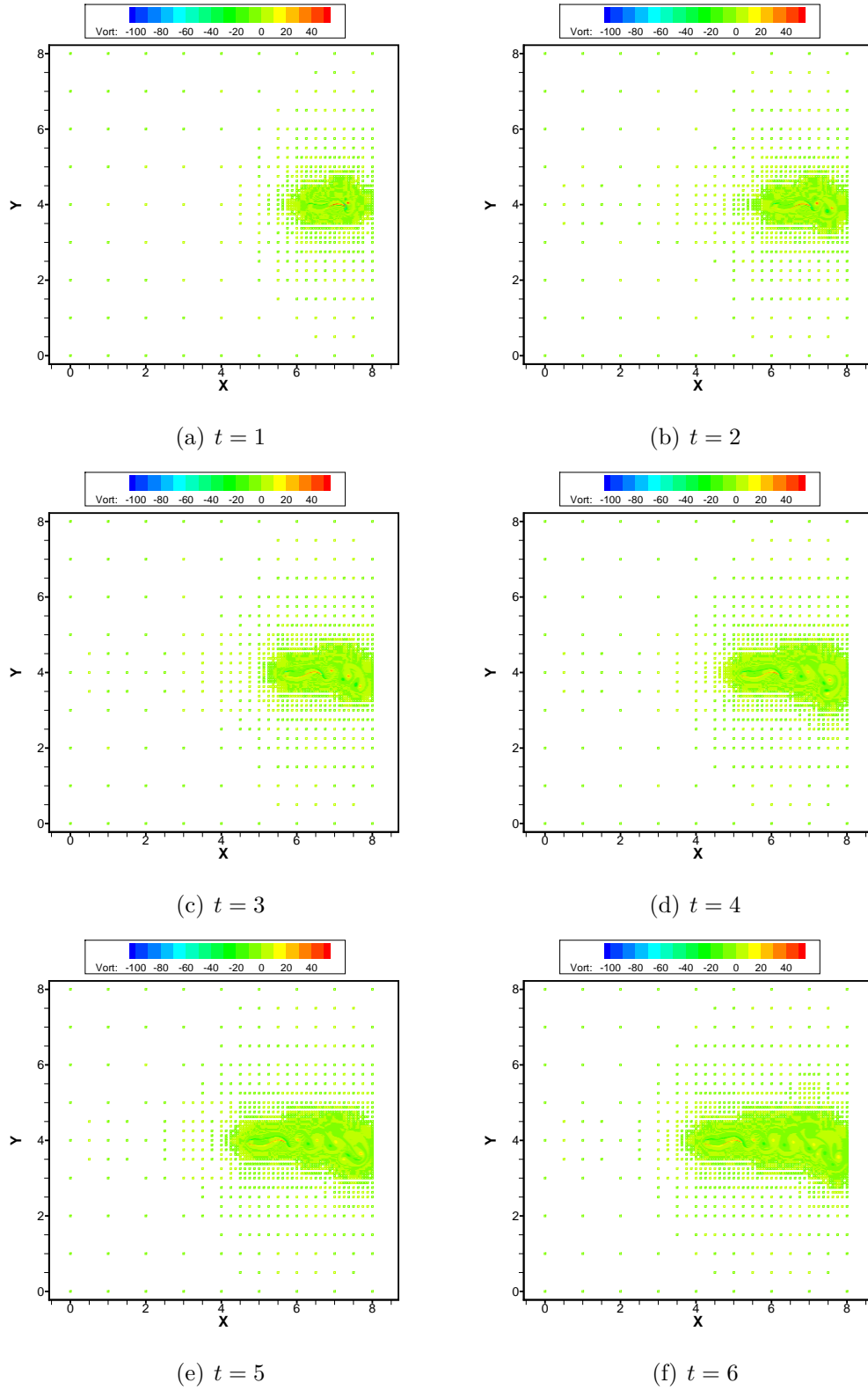


Figure 5.7: Adaptive grids at different instances colored by the vorticity of a swimming fish where $(x, y) \in [0, 8l_{\text{fish}}] \times [0, 8l_{\text{fish}}]$. The penalization parameter inside the body equals $\eta = 10^{-3}$, with a maximum grid level of $J = 10$ in each direction and the viscosity is $\nu = 1.4 \times 10^{-4}$.

Chapter 6

Three-dimensional simulations of fish-like swimming

“Turbulence is any chaotic solution to the three dimensional Navier–Stokes equations that is sensitive to initial data and which occurs as a result of successive instabilities of laminar flows as a bifurcation parameter is increased through a succession of values.”

J. M. McDonough [114] (2007)

Up to now all simulations of the swimming fish performed in two dimensions. For a simple anguilliform swimmer this simplification is logical. If the shape of the swimmer is complicated, i.e., by considering the appendages with independent movements, a three dimensional swimming is inevitable. For this reason the Incompact3d open access code, is adapted for simulation of incompressible flows interacting with deformable bodies. In this Chapter, some mathematical properties of the three dimensional incompressible Navier–Stokes equations, and then the existence and uniqueness of its numerical (weak/strong) solutions, are reviewed. Then conservation of mass, momentum and energy (in the discrete sense) by the Navier–Stokes equations in the inviscid limit are considered. Advective, divergence, skew-symmetric and rotational forms of the convective acceleration term are presented. Also the necessity, advantages and drawbacks of fully or partially staggered grids are discussed. After that, projection method for the numerical simulation of the unsteady incompressible Navier–Stokes equations is presented. The open source Incompact3d code is described including the modifications we made. Finally, validation of the penalized-incompact3d is performed by simulation of a falling sphere in a quiescent

fluid. Last a preliminary simulation of a three dimensional swimming fish is demonstrated. Bottlenecks and future developments are also discussed in the conclusion.

6.1 Governing equations

The governing equations of incompressible flows are the Navier–Stokes equations. For the unit volume of the fluid, in primitive variables they read

$$\underbrace{\rho_f \left(\underbrace{\frac{\partial \mathbf{u}}{\partial t}}_{\text{Local acceleration}} + \underbrace{(\mathbf{u} \cdot \nabla) \mathbf{u}}_{\text{Convective acceleration}} \right)}_{\text{Inertia per volume}} = \underbrace{-\nabla p}_{\text{Pressure gradient}} + \underbrace{\mu \nabla^2 \mathbf{u}}_{\text{Viscous term}} + \underbrace{\mathbf{F}}_{\text{Body forces}} \quad (6.1)$$

and the continuity equation corresponds to

$$\nabla \cdot \mathbf{u} = 0 \quad , \quad \mathbf{x} \in \Omega \in \mathbb{R}^3 \quad (6.2)$$

where Ω is the spatial domain of interest, given as an open subset of \mathbb{R}^3 , which can be bounded or unbounded in general, $\mathbf{u}(\mathbf{x}, t)$ is the velocity field, $p(\mathbf{x}, t)$ represents the pressure, $\mu = \nu/\rho_f$ is the dynamic viscosity, ρ_f is the density of the fluid and $\mathbf{F}(\mathbf{x}, t)$ is a forcing term for the unit volume of the fluid. Proper initial and boundary conditions must be considered to complete the equations. By choosing respectively U_∞ and L as reference velocity and length for a given problem the Navier–Stokes equations can be written in non-dimensional form in which $Re = U_\infty L/\nu$ is the Reynolds number:

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + Re^{-1} \nabla^2 \mathbf{u} + \mathbf{F} \quad (6.3)$$

An equation for the pressure can be derived, by applying the divergence operator to the momentum equations and making use of the continuity:

$$\nabla \cdot (\nabla p) = -\rho_f \nabla \cdot \left((\mathbf{u} \cdot \nabla) \mathbf{u} \right) + \nabla \cdot \mathbf{F} \quad (6.4)$$

The following boundary conditions can be used under the assumption of high Reynolds flow

$$\frac{\partial p}{\partial \mathbf{n}} = 0 \quad \text{on } \partial\Omega \quad (6.5)$$

The Navier–Stokes equations can be expressed in tensor (Einstein's) notation as follows:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (6.6)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho_f} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + f_i \quad i = 1, 2, 3 \quad (6.7)$$

6.1.1 Mathematical properties of the N-S equations

Some mathematical aspects of the Navier–Stokes equations that have an impact on numerical simulations are reviewed from [115].

1. The N-S equations are nonlinear and sensitive to initial conditions. Increasing the Reynolds number results in successive instabilities from laminar flow to periodic, then to quasiperiodic and finally to a chaotic flow.
2. Space ($\mathbf{x}' = \mathbf{x} + \mathbf{x}_0$, $\mathbf{x}_0 \in \mathbb{R}^3$) and time ($t' = t + t_0$, $t_0 \in \mathbb{R}$) translations are symmetry groups of the N-S equations, i.e., just derivatives with respect to (\mathbf{x}, t) appear in the N-S equations.
3. The N-S equations are Galilean invariant ($\mathbf{x}' = \mathbf{x} + \mathbf{u}_0 t$, $\mathbf{u}' = \mathbf{u} + \mathbf{u}_0$, $\mathbf{u}_0 \in \mathbb{R}^3$), thus small scale fluid flow experiments in different parts of the world lead to the same results.
4. The N-S equations describe a dissipative flow which is thus irreversible from thermodynamics viewpoint, but the Euler equations ($\nu = 0$) describe an isentropic reversible flow.
5. For two dimensional flows, existence and uniqueness of weak and strong solutions to the N-S equations have been proven for all times.
6. For three dimensional flows, long-time existence can be demonstrated for weak solutions, but uniqueness has not been proven for this case. On the other hand, only short-time existence has been proven for 3D strong solutions, but it is known that these are unique.

With regard to the existence of strong solutions, some constraints have to be imposed on the shape of the domain (in particular smoothness of $\partial\Omega$), the boundary and initial conditions, the Reynolds number (to be low) and on the body-force terms. We recall that a weak solution is one that is not sufficiently differentiable to be substituted into the differential form of the equations, and instead only satisfies an integral (weak) form of the equations. A strong solution is one that is sufficiently smooth to satisfy the original differential equation in the sense of L^2 .

6.1.2 Conservation properties

It is particularly important, in direct and large eddy simulations of transitional and turbulent flows, that the numerical scheme preserves the conservation properties of the Navier–Stokes equations [81]. In three dimensional flows, the energy goes from the large eddies toward the small eddies via the vortex stretching mechanism (nonlinear convection). If the resolution

of the grid is insufficient, short wave numerical instabilities occur, since the molecular viscosity ν is not sufficient on the grid scales. In general refining the grid solves this problem, but in three dimensions the resolution of the grid is always restricted. Thus a common practice to eliminate the oscillations is to add a kind of numerical dissipation (by high-order filtering or upwinding) or using staggered grids. In the inviscid limit ($Re \rightarrow \infty$), the Navier–Stokes equations reduce to Euler equations. Euler equations conserve mass, linear and angular momentum, energy, circulation, vorticity in two dimensions and helicity in three dimensions in the interior of the flow field. However, numerical integration in time has a damping effect. Periodicity assumption is also necessary because the integral of these quantities over the computational domain can only be affected through the boundaries. In this regards, most numerical schemes do not preserve all these properties. Different representations of the convective term is also affects the conservation properties of the Navier–Stokes equations in the discrete sense. The convective term in the momentum equations can be cast in several ways:

Advective form:

$$(\mathbf{u} \cdot \nabla)u_i = \mathbf{u} \cdot (\nabla u_i) \quad i = 1, 2, 3 \quad (6.8)$$

Where $\nabla = \partial_x \hat{i} + \partial_y \hat{j} + \partial_z \hat{k}$ is the gradient operator and (\cdot) stands for inner product.

Divergence form:

$$\nabla \cdot (u_i \mathbf{u}) \quad i = 1, 2, 3$$

It is obtained with the use of continuity equation and the following relations:

$$\nabla \cdot (\phi \mathbf{A}) = (\nabla \phi) \cdot \mathbf{A} + \phi(\nabla \cdot \mathbf{A})$$

$$(\nabla \phi) \cdot \mathbf{A} = \mathbf{A} \cdot (\nabla \phi) = (\mathbf{A} \cdot \nabla) \phi$$

where \mathbf{A} is a vector, ϕ is a scalar and $\nabla \cdot () = \partial_x() + \partial_y() + \partial_z()$ is the divergence operator. The divergence form can be written in the vector form as $\nabla \cdot (\mathbf{u} \otimes \mathbf{u})$, where \otimes is a special case of tensor product.

Skew-symmetric form:

$$\frac{1}{2} \left((\mathbf{u} \cdot \nabla)u_i + \nabla \cdot (u_i \mathbf{u}) \right) \quad i = 1, 2, 3 \quad (6.9)$$

This is the average of two previous forms. The term skew-symmetric is used because the operator $\frac{1}{2}((\mathbf{v} \cdot \nabla)\mathbf{u} + \nabla \cdot (\mathbf{v}\mathbf{u}))$ is skew-symmetric for fixed \mathbf{v} satisfying $\nabla \cdot \mathbf{v} = 0$.

Rotational form:

$$\omega \times \mathbf{u} + \nabla \left(\frac{|\mathbf{u}|^2}{2} \right)$$

In the Navier–Stokes equations the static pressure can be replaced by the total pressure, $P = p + \frac{|\mathbf{u}|^2}{2}$, to produce the customary rotational version of the momentum equation.

$$\partial_t \mathbf{u} + \omega \times \mathbf{u} + \nabla P = Re^{-1} \nabla^2 \mathbf{u} + \mathbf{F} \quad (6.10)$$

where $\omega = \nabla \times \mathbf{u}$ is the vorticity vector. It is shown by Morinishi et al. in [69] that, if a typical collocated finite difference scheme is used, the advective form does not conserve neither momentum nor energy, the divergence form conserves momentum but not energy, the skew-symmetric and rotational forms conserve both. On the other hand, if a control-volume approach is used, the divergence form conserves energy but the pressure-gradient term does not. With a staggered grid and central differences the conservation properties of the Navier–Stokes equations are preserved. Upwind schemes have undesirable effects on the conservation properties of the method. The same artefact happens by adding explicitly some kind of artificial dissipation. For DNS and LES of incompressible flows, only high-order upwind methods is recommended for discretization of the convective terms. The computational cost of the advective and divergence forms are roughly the same, the rotational form is the cheapest and the skew-symmetric form is the most expensive one.

6.1.3 Grid arrangement

Numerical simulation of unsteady incompressible flows by solving the Navier–Stokes equations requires a method that can accurately represent a wide range of spatial scales. It must be able to produce physical pressure and velocity fields (stable) in addition to good discrete conservation properties. One way to achieve a desired accuracy is to use high-order finite difference schemes for spatial discretization. However, additional constraints such as discrete conservation of mass, momentum and kinetic energy (in the inviscid limit) should be considered (by keeping the time-integration error near the machine precision) if one wants to ensure that unsteady flow simulations are both stable and free of numerical dissipation [80]. The equation for kinetic energy is derived from the momentum equation; it is therefore a consequence of the discretized momentum balance rather than a separate equation. For this reason, the conservation of kinetic energy is commonly referred to as secondary conservation, in contrast to the primary conservation of mass and momentum. A primary reason for probing the conservation of energy in the inviscid limit is that the kinetic energy is a L_2 -norm of the velocity field. A method that conserves this property is guaranteed to be stable against blow-up phenomena. Secondly, it is well-known that absence of artificial dissipation leads to vastly improved accuracy in large eddy simulations where the added sub-grid viscosity is not always positive, thus one cannot rely on the stabilizing effect of

the subgrid model.

The discrete conservation properties of numerical methods also depend on the way flow variables (p, u, v, w) are arranged on the grid. The first choice is the collocated grids where all flow variables are located at the same points which is more advantageous in complex solution domains in comparison with staggered grid in which the flow variables are shifted. On collocated grids, however, the use of symmetric central difference operators give rise to the problem commonly known as pressure checker-boarding. Various methods have been proposed in the literature to overcome this problem when using collocated grids, all of them introduce an explicit or implicit numerical dissipation via central high-order filters, upwinding or introducing a face velocity that depends on an interpolated pressure gradient. As a result, none of them does not conserve kinetic energy. Pressure checker-boarding is due to the resulting wider stencil of the Laplacian in the pressure equation, in which the nearby grid points are decoupled [138]. The discrete Poisson equation for pressure is derived by applying the discrete divergence operator to the momentum equations and making use of the continuity,

$$\frac{\delta}{\delta x_i} \left(\frac{\delta P}{\delta x_i} \right) = -\frac{1}{2} \frac{\delta}{\delta x_i} \left(\frac{\delta u_i u_j}{\delta x_j} + u_j \frac{\delta u_i}{\delta x_j} \right) + \nabla \cdot \mathbf{F} \quad (6.11)$$

In one dimension the discrete Laplacian is the product of two discrete first derivatives. Therefore the discrete Laplacian, without the force term, by using a second order central difference leads to

$$\frac{\delta}{\delta x} \left(\frac{\delta P}{\delta x} \right) = \frac{1}{2\Delta x} \left(\frac{P_{i+2} - P_i}{2\Delta x} - \frac{P_i - P_{i-2}}{2\Delta x} \right) = \frac{P_{i+2} - 2P_i + P_{i-2}}{(2\Delta x)^2} \quad (6.12)$$

which is simply an approximation of the second derivative on a twice coarser grid (higher-order schemes yield a similar result). This implies that its null space includes odd-even oscillations (the π -mode) in addition to constants. In three dimensions, the null space trivially includes all modes that are either constant or odd-even oscillations in three principal directions, i.e., (x) , (y) , (z) and four diagonal directions, i.e., between $(x \& y)$, $(y \& z)$, $(x \& z)$ and $(x \& y \& z)$. Therefore, the null space of the discrete three dimensional Laplace operator is spanned by 8 modes. Elimination of these modes will result in a smooth pressure field without affecting the discrete conservation properties of the Navier–Stokes equations. An attempt in this regard is the work of Shashank et al. [138] in which filtering of the pressure field is restricted to the null space of the Laplacian operator to construct a collocated conservative method. They first solve the discrete Poisson equation and then modify the pressure field by adding some combination of null space modes to produce a smooth pressure field. This is not different from the standard practice in incompressible flow solvers in which an arbitrary mean pressure can be added to the solution. It is a reflection of the fact that the Laplacian operator with Neumann boundary conditions is singular. The method of Shashank et al. [138] is a special kind of filtering of the pressure field, where the filtering

is restricted to the null space of the Laplacian operator to preserve kinetic energy in an inviscid limit. Finally, the common practice of adding dissipation to the pressure equation can be seen as a way to modify the discrete Poisson equation, like in Rhie and Chow [39], but these methods destroy the kinetic energy conservation property.

On staggered grids the flow variables are fully (full-staggered) or partially (half-staggered) shifted. Discrete operators based on central differences with primary and secondary conservation properties have been constructed in several ways. For discretization of the incompressible Navier–Stokes equations over a full-staggered grid, we refer to the pioneering work of Harlow and Welch [20] in which the marker and cell method is used for simulation of incompressible flows in the presence of free surfaces. A half-staggered grid for the pressure is used by Laizet and Lamballais [129] in conjunction with a compact high order discretization of the incompressible Navier–Stokes equations. In Fig. 6.1 a half-staggered grid for the pressure is demonstrated. We have developed a second-order two-dimensional incompressible flow solver based on projection method using half staggered grid for better understanding the difficulties of primitive variables for three dimensional extensions. But in this report, all two dimensional simulations were performed using the vorticity-stream function formulation instead of primitive variables. By using high order discretizations and collocated grids, the vorticity-stream function formulation proved to be more efficient than primitive variables. For three dimensional simulations we are using the Incompact3d open access code developed by Laizet and Lamballais [129]. Some essential aspects of their method will be explained in some details in the following sections.

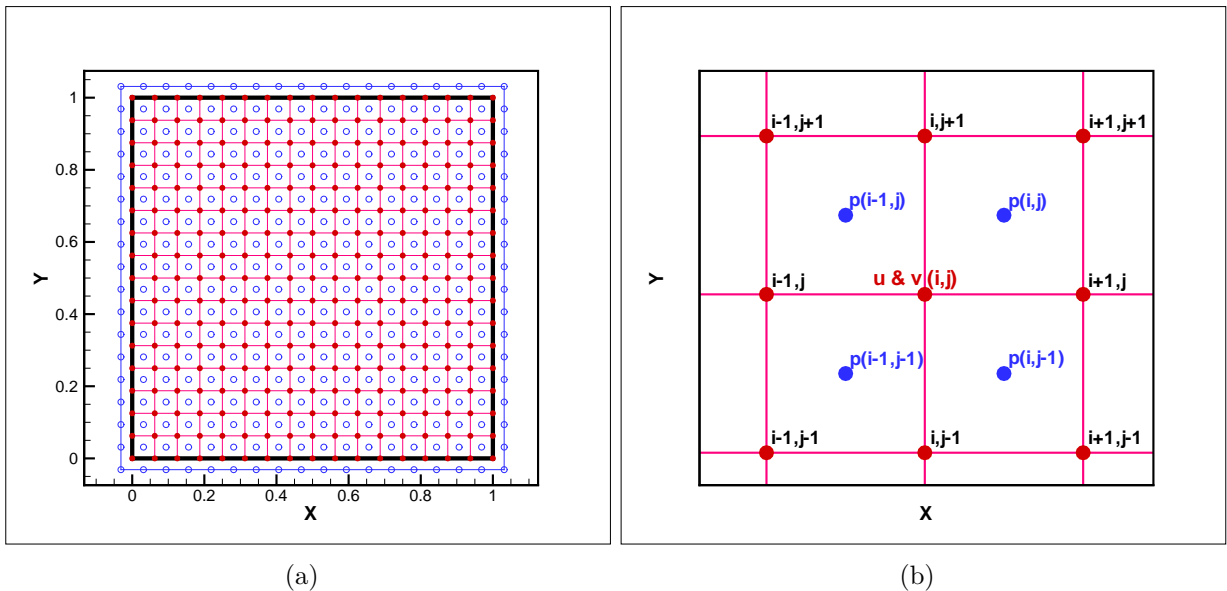


Figure 6.1: (a) Arrangement of velocity \bullet and pressure \circ grids (shifted in x and y directions by $\Delta x/2$ and $\Delta y/2$) in a two-dimensional half-staggered arrangement, physical boundaries are represented by black lines (—), ghost nodes for pressure are on the blue lines ($\circ - \circ - \circ$), (b) Indices of velocity \bullet and pressure \bullet nodes in a two-dimensional half-staggered grid.

6.2 Projection method

The projection method is an efficient approach for numerically solving time-dependent incompressible Navier–Stokes equations. It was originally introduced by Chorin [22] and independently by Temam [23] in 1967. Afterward some improvements were introduced by many other researchers and then commonly used in DNS and LES calculations. Another approach is the artificial compressibility method introduced by Chorin [21], developed by Rogers and Kwak [48] and then by Malan and Lewis [82] and others. In this approach the incompressibility constraint is relaxed by adding the time derivative of pressure field to the continuity equation, thus allowing for pressure correction. The main advantage of projection method is that the computations of the velocity and the pressure fields are decoupled. The algorithm of the projection method is based on the Helmholtz-Hodge decomposition of any vector field into a solenoidal (divergence-free) part and an irrotational part

$$\mathbf{u} = \mathbf{u}_{sol} + \mathbf{u}_{irott} = \mathbf{u}_{sol} + \nabla\phi \quad (6.13)$$

where ϕ is a scalar ($\nabla \times \nabla\phi = 0$) potential function for the irrotational velocity $\mathbf{u}_{irott} = \nabla\phi$. For the solenoidal velocity in two dimensional flows a stream-function ψ can be defined as $\mathbf{u}_{sol} = -\nabla^\perp\psi$. Taking the divergence of Eq. (6.13) by considering $\nabla \cdot \mathbf{u}_{sol} = 0$, yields to a Poisson equation ($\nabla \cdot \mathbf{u} = \nabla^2\phi$) for the scalar function ϕ . If the vector field \mathbf{u} is known, the above equation can be solved for the scalar function ϕ with proper boundary conditions. The solenoidal part of \mathbf{u} can be extracted using the relation

$$\mathbf{u}_{sol} = \mathbf{u} - \nabla\phi \quad (6.14)$$

This is the essence of the projection method for solving the incompressible Navier–Stokes equations. Typically, the algorithm consists of two stages at each time step. In the first stage, an intermediate velocity that does not satisfy the incompressibility constraint is computed by Burgers type equations, i.e., the momentum equation in which the pressure gradient term is dropped. In the second stage, the updated pressure is computed by solving the Poisson equation, in which the predicted velocity field supplies the source term. Finally the updated pressure is used to project the intermediate velocity onto a divergence-free velocity field. For this purpose the Navier–Stokes equations (6.3) can be written as

$$\partial_t \mathbf{u} = -\nabla p - \mathcal{N} + \mathcal{L} + \mathbf{F} = -\nabla p + \mathcal{H} + \mathbf{F} = -\nabla p + \mathcal{R} \quad (6.15)$$

where \mathcal{N} contains the nonlinear terms in one of the previously mentioned forms, \mathcal{L} contains the linear viscous term and F is the forcing term. If we assume that the variation of \mathbf{u} in the time integration is due to $-\nabla p$ and \mathcal{R} , the evolution of the velocity field can be achieved in two separated steps. By defining an intermediate (provisional) velocity field,

say $\hat{\mathbf{u}}$, which is not divergent-free, we have

$$\frac{\hat{\mathbf{u}} - \mathbf{u}^n}{\Delta t} = \mathcal{R} \quad (6.16)$$

and introducing a scalar ϕ as a pseudo pressure, we get

$$\frac{\mathbf{u}^{n+1} - \hat{\mathbf{u}}}{\Delta t} = -\nabla\phi \quad (6.17)$$

The final velocity field \mathbf{u}^{n+1} , which is divergence-free, is obtained by applying a correction to the intermediate velocity field, using the gradient of the scalar ϕ . To find the pseudo pressure, an elliptic equation with Neumann boundary condition must be solved. By taking the divergence of the Eq. (6.17) and enforcing the divergence-free condition for the velocity at time step $(n + 1)$ we obtain

$$\frac{\nabla \cdot \hat{\mathbf{u}}}{\Delta t} = \nabla \cdot (\nabla\phi) \quad (6.18)$$

The boundary condition for ϕ on the domain boundaries $\partial\Omega$ is $\nabla\phi \cdot \mathbf{n} = 0$. If $\mathbf{u} \cdot \mathbf{n} = 0$ on $\partial\Omega$ is prescribed, then the space of divergence-free vector field will be orthogonal to the space of irrotational vector fields, and from equation (6.17) one has

$$\frac{\partial\phi}{\partial\mathbf{n}} = 0 \quad \text{on} \quad \partial\Omega \quad (6.19)$$

The explicit treatment of the boundary condition may be circumvented by using a staggered grid and requiring that $\nabla \cdot \mathbf{u}^{n+1}$ vanishes at the pressure nodes that are adjacent to the boundaries. The boundary conditions for the intermediate velocity field are a challenge, according to Kim and Moin [44] the following relation can be used at the boundaries:

$$\hat{\mathbf{u}} = \mathbf{u}^{n+1} + \Delta t \nabla\phi \quad \text{on} \quad \partial\Omega \quad (6.20)$$

This leads to a slip velocity for the intermediate velocity vector at solid surfaces.

In summary the basic idea of the projection method consists in decoupling the evaluation of the pressure from the velocity components in three complementary steps:

1. First the system is advanced in time to a mid-time-step position, using a suitable advection method in time. This is denoted as the predictor step:

$$\hat{u}_i = u_i^n + \Delta t \mathcal{R}_i^n \quad (i = 1, 2, 3) \quad (6.21)$$

2. Next the elliptic Poisson equation must be solved:

$$D \cdot \mathbf{G}\phi = \frac{D \cdot \hat{\mathbf{u}}}{\Delta t} \quad (6.22)$$

3. Finally the velocity correction (Leray projection) is done:

$$u_i^{n+1} = \hat{u}_i - \Delta t \mathbf{G} \phi \quad (6.23)$$

where D and \mathbf{G} are discrete divergence and gradient operators. The pseudo pressure has a systematic difference with the physical pressure. For the physical pressure Eq. (6.4) must be solved. This stage is optional during the time marching if the physical pressure is not necessary. For three-dimensional simulations we are using the Incompact3d open access code, developed by Laizet and Lamballais [129], in which Adams–Bashforth and Runge–Kutta methods are used for velocity projection.

Explicit Adams–Bashforth methods

The classical projection scheme is first-order accurate in time for the velocity and the pressure field. However second-order accuracy in time for the velocity field can be achieved by using Adams–Bashforth scheme. For this reason, the intermediate velocity prediction (6.21) must be replaced with:

$$\hat{u}_i = u_i^n + \Delta t \left\{ \frac{3}{2} \mathcal{R}_i^n - \frac{1}{2} \mathcal{R}_i^{n-1} \right\} \quad (6.24)$$

Third-order accuracy in time for the velocity field can also be obtained by using third order Adams–Bashforth (AB3) scheme. In this case, the intermediate velocity can be predicted with:

$$\hat{u}_i = u_i^n + \Delta t \left\{ \frac{23}{12} \mathcal{R}_i^n - \frac{16}{12} \mathcal{R}_i^{n-1} + \frac{5}{12} \mathcal{R}_i^{n-2} \right\} \quad (6.25)$$

In high order Adams–Bashforth methods (in contrast to Runge–Kutta methods) smaller Δt must be used, however only one evaluation of the nonlinear convection terms is necessary per time step. In low Reynolds number flows, dominated by the viscous effects, the implicit Crank–Nicolson scheme is recommended for advancing the viscous terms \mathcal{L} , the Adams–Bashforth method can be used for the convective terms \mathcal{N} . Semi-implicit schemes can lead to second-order accuracy in time, they are indeed more robust but at the same time more complicated. Another method proposed by Kim and Moin [44] to integrate implicitly just normal to the wall terms, either convection or diffusion. This method lead to a tridiagonal linear system of equations, that can be solved efficiently by direct methods presented in Appendix F.

Explicit third-order Runge–Kutta method

We present a low-storage third-order Runge–Kutta (RK3) time advancement scheme, where all terms in the RHS of momentum equations are advanced explicitly:

$$\hat{u}_i^k = u_i^{k-1} + \Delta t \{-\gamma_k \mathcal{H}_i^{k-1} + \beta_k \mathcal{H}_i^{k-2} - \alpha_k \nabla p^{k-1} + F_i^k\} \quad (6.26)$$

$$\nabla^2 \phi^k = \frac{\nabla \cdot \hat{\mathbf{u}}_i^k}{\alpha_k \Delta t} \quad (6.27)$$

$$u_i^k = \hat{u}_i^k - \alpha_k \Delta t \nabla \phi^k \quad (6.28)$$

$$p^k = p^{k-1} + \phi^k \quad (6.29)$$

where $k = 1, 2, 3$ is the substep index, \hat{u}_i^k are the intermediate velocities and ϕ is the scalar for velocity correction. Here \mathcal{H} is a spatial operator containing the convective and viscous terms, F_i^k is the momentum forcing term. The coefficients of the third-order Runge–Kutta method are given in Table 6.1. For more details about the projection method we refer to Guermond et al. [108].

Table 6.1: Coefficients of the third-order Runge–Kutta method.

	α_k	β_k	γ_k
$k = 1$	8/15	0	8/15
$k = 2$	2/15	-17/60	5/12
$k = 3$	1/3	-5/12	3/4

6.3 Incompact3d code and our modification

According to Laizet and Lamballais [129] Incompact3d is a powerful numerical tool for academic research. It combines the versatility of industrial codes with the accuracy of spectral codes. It uses a Cartesian mesh which offers the opportunity to implement high-order compact schemes for the spatial discretization while an immersed boundary method (IBM) allows the implementation of any complex solid body inside the computational domain. To ensure the incompressibility constraint, the Poisson equation is fully solved in spectral space via the modified wave number formalism, no matter what the boundary conditions are (periodic, free-slip, no-slip, inflow/outflow, etc.). The pressure grid is staggered from the velocity grid by half grid spacing in each direction (see Fig. 6.1) to avoid spurious pressure oscillations. Introducing a solid body in the solutions domain may cause additional oscillations in the pressure field near the immersed boundaries [129]. The combination of high-order schemes with IBM can be problematic because of the discontinuity in velocity

derivatives, locally resulted from the forcing term. Even though, the formal order of the solution can be reduced as a result of the IBM, the code has been demonstrated to be far more accurate with a 6^{th} -order scheme than with a second order scheme both in statistics and instantaneous field realizations [125]. Incompact3d can be used on up to hundreds of thousands computational cores to solve the incompressible Navier–Stokes equations [132]. This high level of parallelization is achieved thanks to a highly scalable two dimensional decomposition (see Fig. 6.2) library and a distributed FFT interface [142] which is available on-line (2DECOMP & FFT).

DNS/LES solutions are frequently based on high-order schemes free from dissipation error through the use of a centered formulation over structured uniform grids. The advantage of a 6^{th} -order compact finite difference scheme proposed by Lele [55] over an explicit second order method is illustrated in Fig. 6.3 (d) in terms of scaled modified wavenumber for the first derivative. However, the lack of accuracy at small scales combined with aliasing errors, the error in boundary conditions, failure of conservation properties at the discrete level, using immersed boundary methods, ..., frequently lead to spurious high wavenumber oscillations at grid spacing scales. These oscillations can be controlled by the physical dissipation using a highly refined mesh, at the expense of increasing computational effort. By using marginal resolutions to reduce the cost of the computations, various techniques are commonly used to suppress or reduce the spurious oscillations. According to Lamballais et al. [150] the robustness of the computational algorithm can be improved by a relevant choice for the discretization of the governing equations, in order to ensure conservation properties, or through the mesh arrangement (grid staggering, see Section 6.1.3). One of the most popular methods to control the spurious oscillations is to use upwind schemes to compute the convective terms in order to reinforce numerical dissipation near the mesh cutoff wavenumber. The mesh cutoff wavenumber also called the π -mode is illustrated in Fig. 6.3 (d). A similar effect can be obtained using a specific artificial damping term, or a filtering procedure proposed by Visbal and Gaitonde in [85]. Upwinding, damping or filtering techniques are essentially non-conservative methods that introduce explicitly some numerical dissipation. In implicit LES, artificial dissipation can be interpreted as a subgrid model ensured by upwinding, damping or filtering.

In the Incompact3d code, a simple scheme is used to introduce some numerical dissipation without the use of any upwinding, damping or filtering operator. The extra dissipation is directly enclosed in the viscous terms of the Navier–Stokes equations through a manipulated 6^{th} -order compact finite difference scheme proposed by Lamballais et al. in [150] for the computation of second derivatives. The effect of manipulation of the coefficients of the classical 6^{th} -order compact scheme is illustrated in Fig. 6.3 (a) - (b) in terms of the scaled modified wavenumber beside the exact differentiation and the original 6^{th} -order compact scheme proposed by Lele [55]. The extra dissipation obtained at small scales for compact schemes is used to freely adjust the level of numerical dissipation near the mesh cutoff

wavenumber while ensuring high-accuracy, i.e., the method is almost free from any numerical dissipation at large scales. The spectral property of the proposed scheme is compared in Fig. 6.3 (e) with a previously proposed high-order upwind approach. In the context of spectral methods, hyperviscosity can be used to artificially extend the inertial range in a turbulent flow while ensuring numerical dissipation near the mesh cutoff. In Fig. 6.3 (f) comparison of the proposed method with hyperviscosity is represented. In the same spirit, the spectral vanishing viscosity method can also be viewed as an alternative LES model which can affect the smallest scales without adding extra dissipation at large scales. The comparison of the proposed scheme with a spectral vanishing viscosity is illustrated in Fig. 6.3 (c). For more information we refer to Lamballais et al. [150] and the cited references. In the light of these illustrations the solutions of the Incompact3d code can be classified to be implicit LES by using a coarse grid, rather than a DNS over a fine grid.

The possible boundary conditions in the Incompact3d code are periodic, free-slip and Dirichlet. In Table 6.2 some necessary informations regarding the possible numbers for each type of boundary condition are listed. In the code velocity grid resolution (nx, ny, nz) , pressure grid resolution (nxm, nym, nzm) , column-wise and row-wise CPU numbers $(prow, pcol)$ for the domain decomposition must be set in the module PARAM. Other informations like time step, time integration method, domain size, boundary conditions $(nclx, ncly, nclz)$, Reynold number, discretization of the convection term (skew-symmetric/rotational), initial noise and etc. are set in the *Incompact3d.prm* input file. We use the skew-symmetric discretization of the convection term in all simulations.

Table 6.2: The possible boundary conditions in the Incompact3d code, (ℓ is power of 2, 3, 4, 5 and 6).

Boundary condition	Periodic	Free-slip	Dirichlet
ncl	0	1	2
n (velocity grid resolution)	2ℓ	$2\ell + 1$	$2\ell + 1$
nm (pressure grid resolution)	n	n-1	n-1

For time discretization several options are available in the code, e.g., second and third order Adams–Bashforth (AB) and also third and fourth order Runge–Kutta (RK) methods. For spatial discretization a 6th-order compact scheme is used. First order direct forcing is used to insert a solid body inside the computational domain. In a direct forcing method, the force in the RHS of the Navier–Stokes equations is defined as:

$$\mathbf{F} = \chi \left(\frac{\mathbf{u}_P - \mathbf{u}}{\Delta t} + \nabla p + \mathcal{N} - \mathcal{L} \right) \quad (6.30)$$

The first-order (in space) application of this method [129] will not give the exerted force on the body. A second-order application via ghost cells [91] can give the exerted forces on the

body by increasing the cost of the computations considerably. In the volume penalization method the force in the RHS of the Navier–Stokes equations is defined as

$$\mathbf{F} = \frac{\chi}{\eta}(\mathbf{u}_P - \mathbf{u}) \quad (6.31)$$

the sum of this term over the volume gives the exerted forces on the body which is an advantage of the volume penalization method. Therefore in the present investigation we prefer to use the volume penalization method. Comparison of the obtained results via explicit and implicit implementation of the volume penalization term in the time integration reveals the advantage of implicit penalization over the explicit one. In the first step (velocity penalization) the velocity of the body is imposed where $\chi \neq 0$:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{\chi}{\eta}(\mathbf{u}_P - \mathbf{u}^n) \quad (6.32)$$

Implicit penalization is used in this step. Then the hydrodynamic coefficients must be evaluated from Eqs. (2.22) and (2.23) for calculation of the displacements and the rotations. Next a velocity prediction is performed via:

$$\frac{\mathbf{u}^{**} - \mathbf{u}^*}{\Delta t} = -\mathcal{N} + \mathcal{L} \quad (6.33)$$

then a Leray projection is done via:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{**}}{\Delta t} = -\nabla\phi \quad (6.34)$$

Finally, for the pseudo pressure the following relation is derived by the author for the implicit penalization-projection method:

$$\nabla^2\phi = \nabla \cdot \left[\frac{\mathbf{u}^{**}}{\Delta t} - \frac{\chi}{\eta}(\mathbf{u}^{**} - \mathbf{u}_P) \right] \quad (6.35)$$

For other choices we refer to Belliarda and Fournier [139]. By setting $\mathbf{u}_P = 0$ and $\eta = \Delta t$, like in the direct forcing method for a fixed body, the Poisson equation proposed by Laizet and Lamballais [129] is recovered as follows:

$$\nabla^2\phi = \nabla \cdot \left[\frac{(1 - \chi)\mathbf{u}^{**}}{\Delta t} \right] \quad (6.36)$$

Setting $\chi = 0$ leads to the Poisson equation (6.22) of the projection method. For the physical pressure, if necessary, Eq. (6.4) must be solved.

6.4 Validation of the algorithm for a falling sphere

In this section the ability of the penalized Incompact3d code in dealing with fluid–structure interaction in three dimensions is examined. The considered test case is a falling sphere due to the gravity, at $Re_d = dU/\nu = 100$. A rigid sphere with $\rho_s > \rho_l$ is released from rest and accelerates until it reaches its asymptotic fall velocity $u_{\text{streamwise}} \approx 1$. The sphere diameter is set to $d = 1.0$ and the kinematic viscosity is chosen as $\nu = 0.01$ to obtain an asymptotic falling velocity $u_{\text{streamwise}} = 1.0$. In $Re_d = 100$, the drag coefficient for the flow past a sphere $C_D = 1.1$ is given by Johnson and Patel [74]. The gravity constant g and ρ_s is determined by Kern and Koumoutsakos [105] using

$$F_{\text{gravity}} - F_{\text{buoyancy}} = F_D = \frac{1}{2}C_D\rho_f U^2(\pi d^2/4)$$

and

$$\rho_s/\rho_l = 1 + C_D(3U^2/4g)$$

By choosing $g = 20$, the density of the sphere must be set to $\rho_s = 1.041$. The grid used by Kern and Koumoutsakos [105] is a body fitted O-O type with radius $r = 15$ and $10 \times 40 \times 100$ cells with exponential clustering towards the wall, in accordance to the reference grid used by Johnson and Patel [74]. The time step was set to $\Delta t = 0.001$ by Kern and Koumoutsakos [105]. Some simulations were performed with the penalized Incompact3d solver, using third-order Adams–Bashforth method with $\Delta t = 10^{-4}$. According to Kern and Koumoutsakos [105] at time $t = 20$ an asymptotic falling velocity of $u_{\text{streamwise}} = 1.006$ is reached, which reasonably matches the predicted value relevant to the chosen parameters. We had observed after $t = 10$ the change in the value of the streamwise velocity is negligible. Therefore for reduction of the domain size, iterations and thus the computational effort we are comparing the results of different simulations up to $t = 10$ with that of Kern and Koumoutsakos [105].

By using different penalization parameters η , the evolution of the streamwise velocities V_x of the falling sphere, are compared with the simulation performed by Kern and Koumoutsakos [105] in Fig. 6.8. The corresponding streamwise force and displacement of the falling sphere for a penalization parameter $\eta = 10^{-2}$ are given in Fig. 6.9. In the streamwise velocity a maximum 10% difference is visible between our simulations and that of Kern and Koumoutsakos [105]. In our opinion this is due to insufficient spatial resolution. Fig. 6.4 shows snapshots of the Q isosurfaces. The z -mid section velocities and streamlines colored by streamwise velocity at different instances are given in Figs. 6.5, 6.6 and 6.7.

6.5 Three dimensional simulation of swimming fish

To be able to carry out a three dimensional simulation of the swimming fish, first of all the orientation and the position of the fish must be described in proper coordinate. As in our two-dimensional simulations we choose the orientation and the position of the head as reference point in all calculations. To this end in three dimensions, we must use the Euler angles beside the coordinates of the head in Cartesian system. We denote the Euler angles by (ϕ_x, ϕ_y, ϕ_z) which are equivalent to roll, pitch and yaw on the body frame in the hydrodynamic literature. Because of some ambiguities and singularities in the Euler angles, following Boyer et al. [106] we use the quaternions for describing the orientation of the head and also calculation of the deformation and kinematics of the backbone of the considered fish. Quaternions were first introduced by Hamilton [1] in 1843 and were then extensively used in many physics and geometry problems. For describing spatial rotations, they have been used in a wide range of applications: computer graphics, optics, robotics, applied mathematics, aerodynamics and orbital mechanics. Quaternions are a non-singular representation of rotation, unlike the Euler angles, even if they are less intuitive than direct angles. Moreover, for describing a rotation, quaternions are favored over trigonometric approaches, because of their remarkably compact form. For more information we refer to Lazarus et al. [154]. To find the quaternion of the head which will be used as initial condition for evaluation of the geometry and kinematics of the backbone we have

$$q_0 = c_1 c_2 c_3 - s_1 s_2 s_3 \quad (6.37)$$

$$q_1 = s_1 s_2 c_3 + c_1 c_2 s_3 \quad (6.38)$$

$$q_2 = s_1 c_2 c_3 + c_1 s_2 s_3 \quad (6.39)$$

$$q_3 = c_1 s_2 c_3 - s_1 c_2 s_3 \quad (6.40)$$

where

$$c_1 = \cos(\phi_y/2) \quad , \quad s_1 = \sin(\phi_y/2)$$

$$c_2 = \cos(\phi_z/2) \quad , \quad s_2 = \sin(\phi_z/2)$$

$$c_3 = \cos(\phi_x/2) \quad , \quad s_3 = \sin(\phi_x/2)$$

The inverse transform is given by

$$\phi_x = \arctan \left(\frac{2q_0 q_1 + 2q_2 q_3}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \right) \quad (6.41)$$

$$\phi_y = \arcsin(2q_0 q_2 - 2q_1 q_3) \quad (6.42)$$

$$\phi_z = \arctan \left(\frac{2q_1 q_2 + 2q_0 q_3}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \right) \quad (6.43)$$

In the FORTAN implementation, *atan2* is used to avoid sign and range mistakes. The range of the rotation angles are

$$\phi_x, \phi_z \in [-\pi, \pi]$$

and

$$\phi_y \in [-\frac{\pi}{2}, \frac{\pi}{2}]$$

A correction must be applied if the range is not in the given interval. A rotational motion of a solid body is governed by the Euler equations of motion. In the inertial frame we have

$$\Sigma \mathbf{M}_{ref}^i = \mathbf{I}_i(t) \vec{\alpha} \quad (6.44)$$

where $\mathbf{I}_i(t)$ is the moment of inertia matrix in an inertial frame which is time dependent (to be determined with respect to the reference point). The definition of the moment of inertia matrix is given in Appendix B. In the body frame the Euler equations are given as follows

$$\Sigma \mathbf{M}_{ref}^b = \mathbf{I}_b \vec{\alpha} + \vec{\omega} \times \mathbf{I}_b \vec{\omega} \quad (6.45)$$

where \mathbf{I}_b is the moment of inertia matrix in a body frame which is constant, to be determined with respect to the reference point once before the simulation. The second term in the right hand side of Eq. (6.45) is the coupling term and must be evaluated by choosing the body frame for representation of the orientation. If the body frame coincides with the principal axes of inertia the moment of inertia matrix will be diagonal for a symmetric shape. The reference point can be the center of gravity. Integration of the Euler equations (6.45) in the body frame gives the angular velocities of the body

$$\frac{d\vec{\omega}}{dt} = \mathbf{I}_b^{-1} (\Sigma \mathbf{M}_{ref}^b - \vec{\omega} \times \mathbf{I}_b \vec{\omega}) \quad (6.46)$$

The change of the Euler angles through the Euler equations will not lead to roll, pitch and yaw angles in a straightforward manner. With the use of the quaternions, the body frame orientation is given by the following relation by Rafei et al. [120]

$$\frac{dQ}{dt} = \frac{1}{2} M^\vee(\vec{\omega}) Q \quad (6.47)$$

where $M^\vee(\vec{\omega})$ is an antisymmetric matrix. After determination of the head orientation, like two-dimensions the geometry and kinematics of the backbone must be determined, see Body kinematics in Algorithm 2. Then geometry of the fish is constructed by a series of ellipses by given height and width normal to the backbone of the fish, see Fig. 6.10. The

width $w(s)$ is given by Gazzola et al. [144] as follows:

$$w(s) = \begin{cases} \sqrt{2w_h s - s^2} & 0 \leq s < s_b \\ w_h - (w_h - w_t)\left(\frac{s-s_b}{s_t-s_b}\right)^2 & s_b \leq s < s_t \\ w_t \frac{L-s}{L-s_t} & s_t \leq s \leq L \end{cases} \quad (6.48)$$

where L is the body length, $w_h = s_b = 0.04L$, $s_t = 0.95L$ and $w_t = 0.01L$. The height $h(s)$ is given by Gazzola et al. [155] as follows:

$$h(s) = \begin{cases} h_1 \sqrt{1 - \left(\frac{s-s_1}{s_1}\right)^2} & 0 \leq s < s_1 \\ h_1 - 2(h_2 - h_1)\left(\frac{s-s_1}{s_2-s_1}\right)^3 + 2(h_2 - h_1)\left(\frac{s-s_1}{s_2-s_1}\right)^2 & s_1 \leq s < s_2 \\ h_2 - 2(h_3 - h_2)\left(\frac{s-s_2}{s_3-s_2}\right)^3 + 3(h_3 - h_2)\left(\frac{s-s_2}{s_3-s_2}\right)^2 & s_2 \leq s < s_3 \\ h_3 \sqrt{1 - \left(\frac{s-s_3}{L-s_3}\right)^2} & s_3 \leq s \leq L \end{cases} \quad (6.49)$$

where $(s_1, h_1) = (0.284L, 0.072L)$, $(s_2, h_2) = (0.844L, 0.041L)$ and $(s_3, h_3) = (0.957L, 0.071L)$. See Fig. 6.11 for the profiles of the fish. On each discrete point, describing the geometry of the backbone, using the given height and width, an ellipse is form at the origin in the $(y - z)$ plane ($x' = 0$) with the following parametric equations:

$$y'(s) = w(s) \sin(t) \quad , \quad z'(s) = h(s) \cos(t) \quad , \quad t \in [0, 2\pi] \quad (6.50)$$

The created ellipse then must be turned and moved to the right position by a quaternion based rotation matrix to be normal to the backbone, see Fig. 6.12 and Fig. 6.13.

$$(x, y, z)^T = Rot(Q)(x', y', z')^T + (x, y, z)_{\text{camber}}^T \quad (6.51)$$

The corresponding velocity components of each point on the Lagrangian grid V_{shape} with (I, J, K) indexes are given by

$$V_{\text{shape}}(I, J, K) = V_{\text{BN}}(I) + \mathbf{r}(I, J, K) \times \Omega_{\text{BN}}(I) \quad (6.52)$$

where V_{BN} and Ω_{BN} are respectively the linear and angular velocities of the backbone, given by Eq. (4.14). The radius $\mathbf{r} = (x, y, z) - (x, y, z)_{\text{camber}}$ must be evaluated at each point of the ellipses. Then with the use of vector product of \mathbf{r} with the angular velocities in the inertial frame, the velocity vector of each point of the Lagrangian grid is calculated by Eq. (6.52). See Fig. 6.14 for an example of the grid covering the fish after deformation and the corresponding velocities of each point. The information of the Lagrangian structured grid covering the deformable body must be transfered to the Eulerian-Cartesian grid by interpolation to find $\chi(i, j, k)$ and $\mathbf{u}_p(i, j, k)$. We use a (three-dimensional) tri-linear

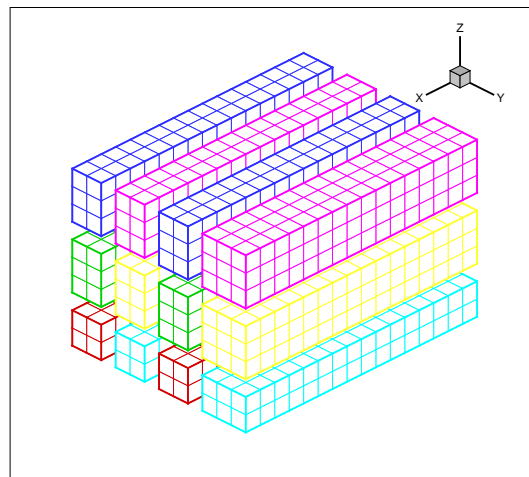
interpolation

$$f(x, y, z) = axyz + bxy + cxz + dyz + ex + fy + gz + h$$

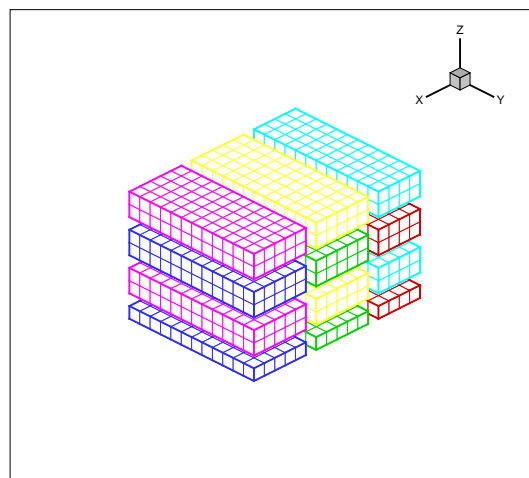
which leads to a 8×8 linear system to find the unknown coefficients. The linear system is solved by the direct Gauss-Jordan elimination method from Numerical Recipes [56]. For each point of the Eulerian grid in which $\chi \neq 0$ the eight nearest points of the Lagrangian grid are used. For some points in which $0 < \chi < 1$ due to mollifying by Eq. (6.53), the interpolation automatically becomes an extrapolation if the point is outside of the original Lagrangian shape. All points in the interior of the fish have $\chi(i, j) = 1$ on the Eulerian grid. The mask is mollified by the Shuman [15] filter (6.53)

$$\bar{\chi}(i, j, k) = (2\chi_{i,j,k} + \chi_{i+1,j,k} + \chi_{i-1,j,k} + \chi_{i,j+1,k} + \chi_{i,j-1,k} + \chi_{i,j,k+1} + \chi_{i,j,k-1})/8 \quad (6.53)$$

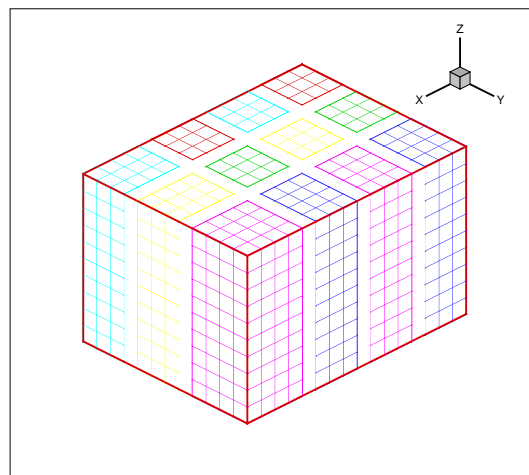
The interpolated mask function ($\bar{\chi}$) and the velocity components over the Eulerian grid are shown in Fig. 6.15. The numbers of grid points on the Lagrangian grid must be fine enough in comparison to the Eulerian grid to accurately represent the deformation of the body. With the developed algorithm, some preliminary simulations of swimming fish are performed. A three dimensional fish with length $l = 1$, tail beat frequency $f = 1$, wavenumber $\lambda = 1$ is swimming in a quiescent fluid. The buoyancy is equal to zero $\rho_b = \rho_f$, the size of the domain is $(x, y, z) \in [0, 3] \times [0, 1] \times [0, 1]$, spatial resolution is $257 \times 101 \times 101$ and the boundary condition according to Table 6.2 are BC (2-1-1). By using third order Adams–Bashforth method in the penalized Incompact3d solver, the time step $\Delta t = 2 \times 10^{-4}$ is used. The penalization parameter is $\eta = 10^{-3}$ and the filter parameter for denoising of the hydrodynamic coefficients is $\delta_{\text{filter}} = 10^{-3}$. The kinematic viscosity is set to $\nu = 2 \times 10^{-3}$. The results of one case for swimming fish at $Re \approx 100$ are reported. The Q iso-surfaces of the swimming fish are illustrated in Fig. 6.17 at different instances. The streamlines colored by streamwise velocity are shown in Fig. 6.18. The z -mid velocity fields are demonstrated in Fig. 6.19. The forces, velocities and the trajectories of the swimming fish are plotted in Fig. 6.20. As can be seen the swimming mechanism in three dimension is more complicated than two-dimension. The essential idea is to create vortical structures by performing a wavy motion by the body in the surrounding flow. As in two-dimensions the created trust is oscillatory.



(a) X-pencils

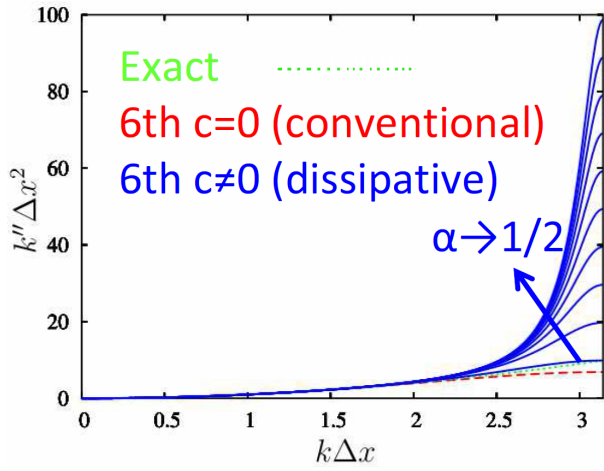
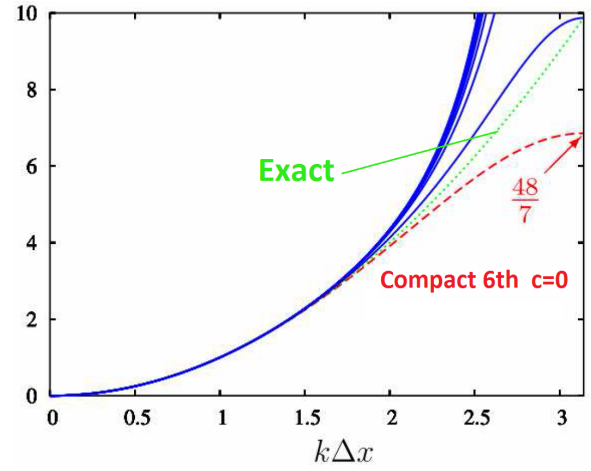
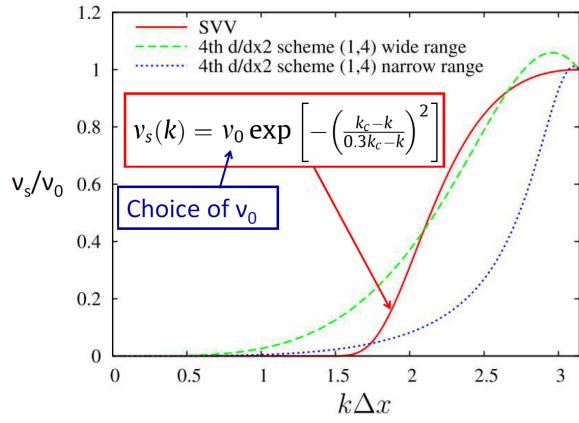


(b) Y-pencils

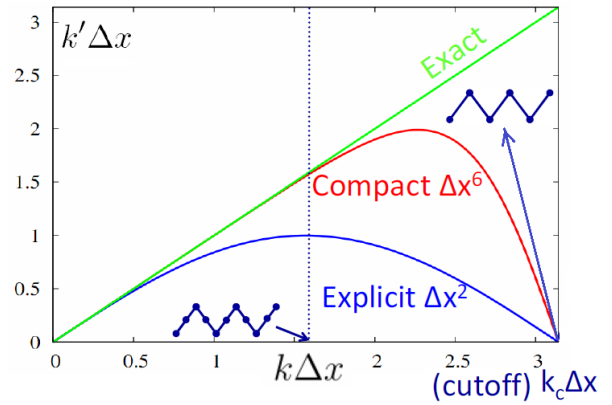
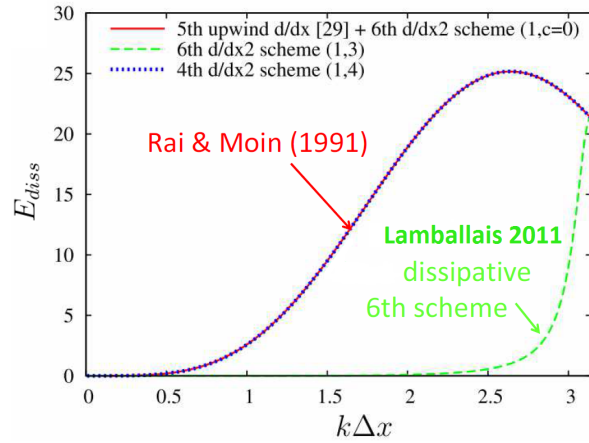


(c) Z-pencils

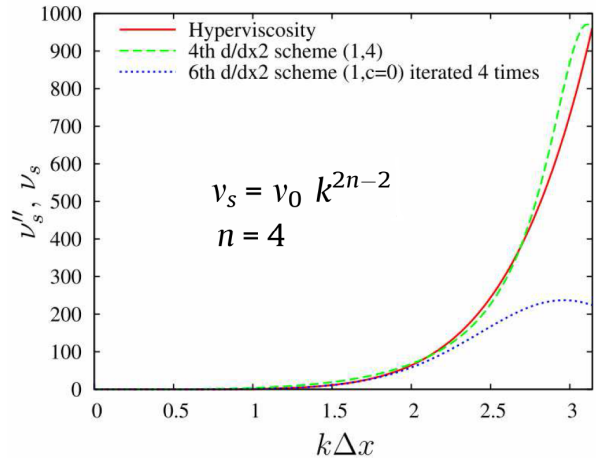
Figure 6.2: An example of two dimensional domain decomposition using 4×3 (row \times column) processors. For data in the X-pencils one global operation in the Z-pencils direction needs 4 data transpositions to come back to the X-pencils, i.e., $X \rightarrow Y \rightarrow Z$ (operation) $\rightarrow Y \rightarrow X$. Pencil rotation (transpose) is done via the MPI “ALL TO ALL” subroutine.

(a) Modified wavenumber for the 2nd derivative(b) Mod. waven. for the 2nd der. (zoom)

(c) Comparison with a spectral vanishing viscosity

(d) Modified wavenumber for the 1st derivative

(e) Comparison with a high-order upwind method



(f) Comparison with a hyperviscosity method

Figure 6.3: Some characteristics of the Incompact3d code in terms of scaled wavenumber in comparison to other methods, pictures are taken from Lamballais et al. [150].

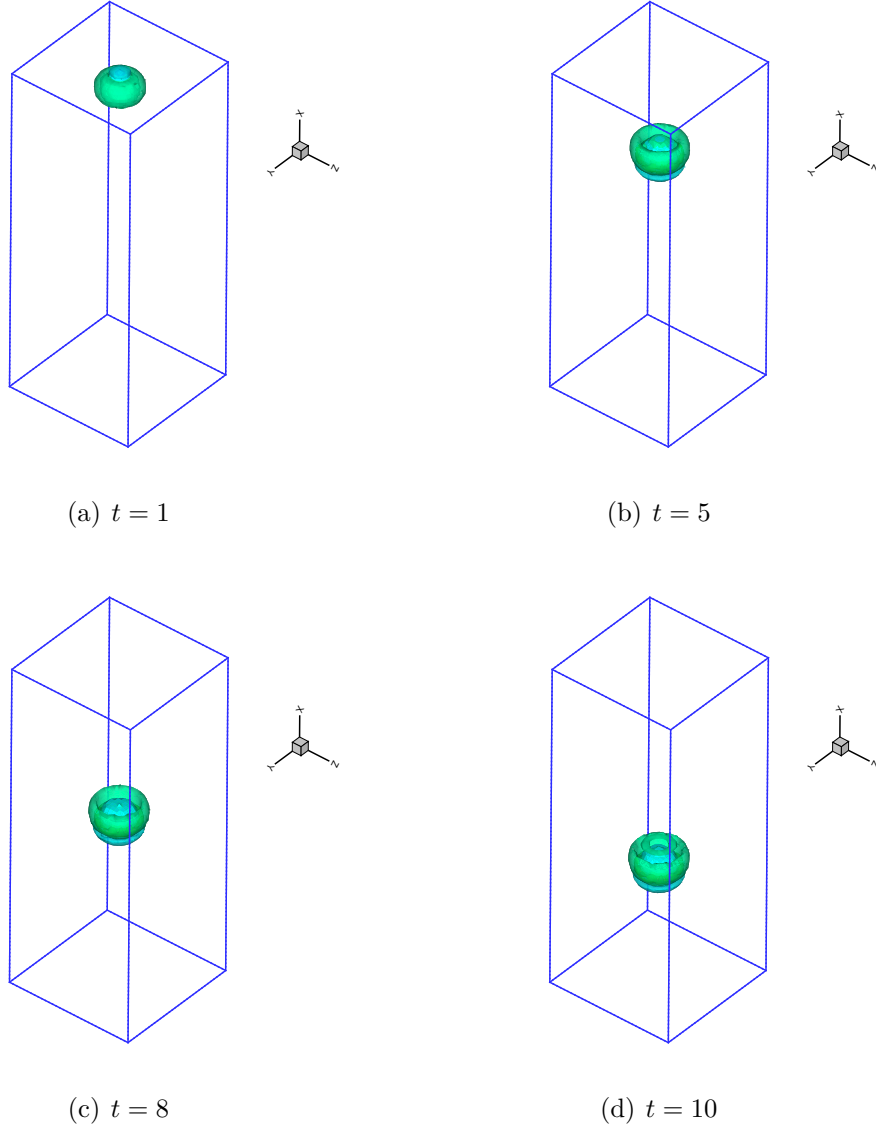


Figure 6.4: The Q isosurfaces of the falling sphere in a quiescent fluid, obtained with the penalized Incompact3d solver, where $g = -20$, $\rho_b/\rho_f = 1.041$, $D = 1$, $(x, y, z) \in [0, 12] \times [0, 4] \times [0, 4]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-2}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 10^{-2}$ and $Re \approx 100$.

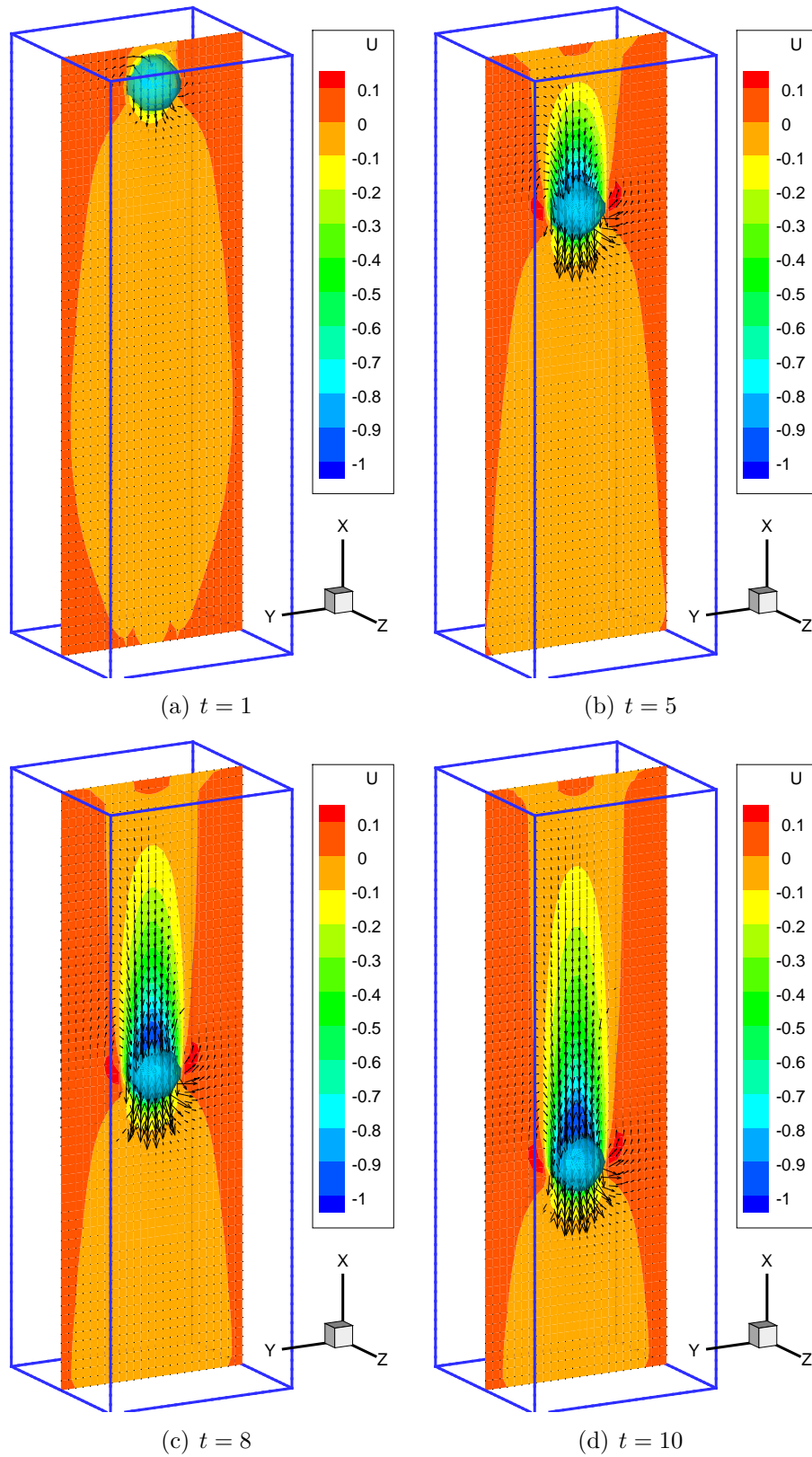


Figure 6.5: The z -mid section velocities of the falling sphere in a quiescent fluid, obtained with the penalized Incompact3d solver, where $g = -20$, $\rho_b/\rho_f = 1.041$, $D = 1$, $(x, y, z) \in [0, 12] \times [0, 4] \times [0, 4]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-2}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 10^{-2}$ and $Re \approx 100$.

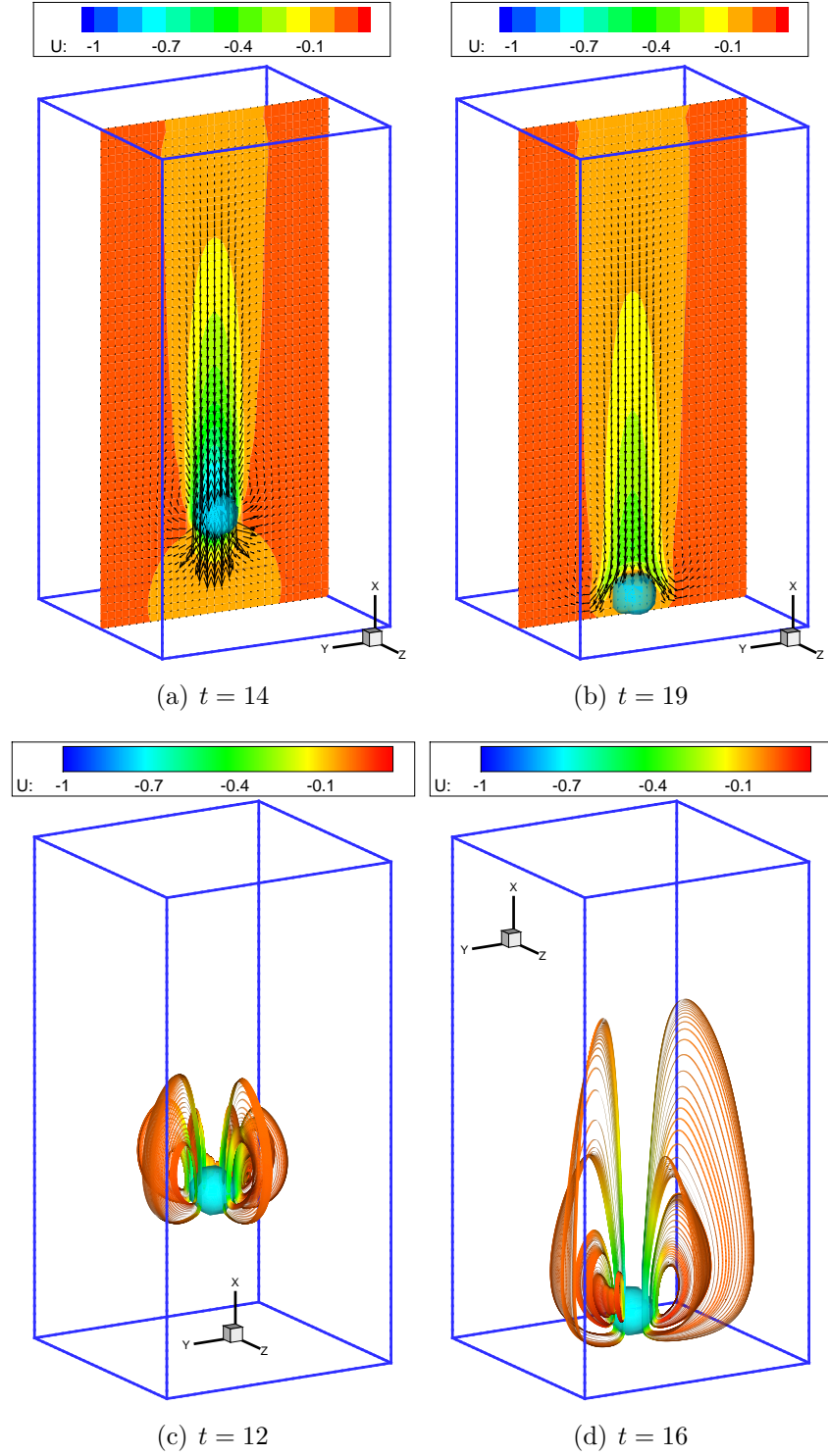


Figure 6.6: The z -mid section velocities (up) and streamlines colored by streamwise velocity (down) of the falling sphere in a quiescent fluid obtained with the penalized Incompact3d solver, where $g = -20$, $\rho_b/\rho_f = 1.041$, $D = 1$, $(x, y, z) \in [0, 12] \times [0, 6] \times [0, 6]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 10^{-3}$ using AB2, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-2}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 10^{-2}$ and $Re \approx 100$.

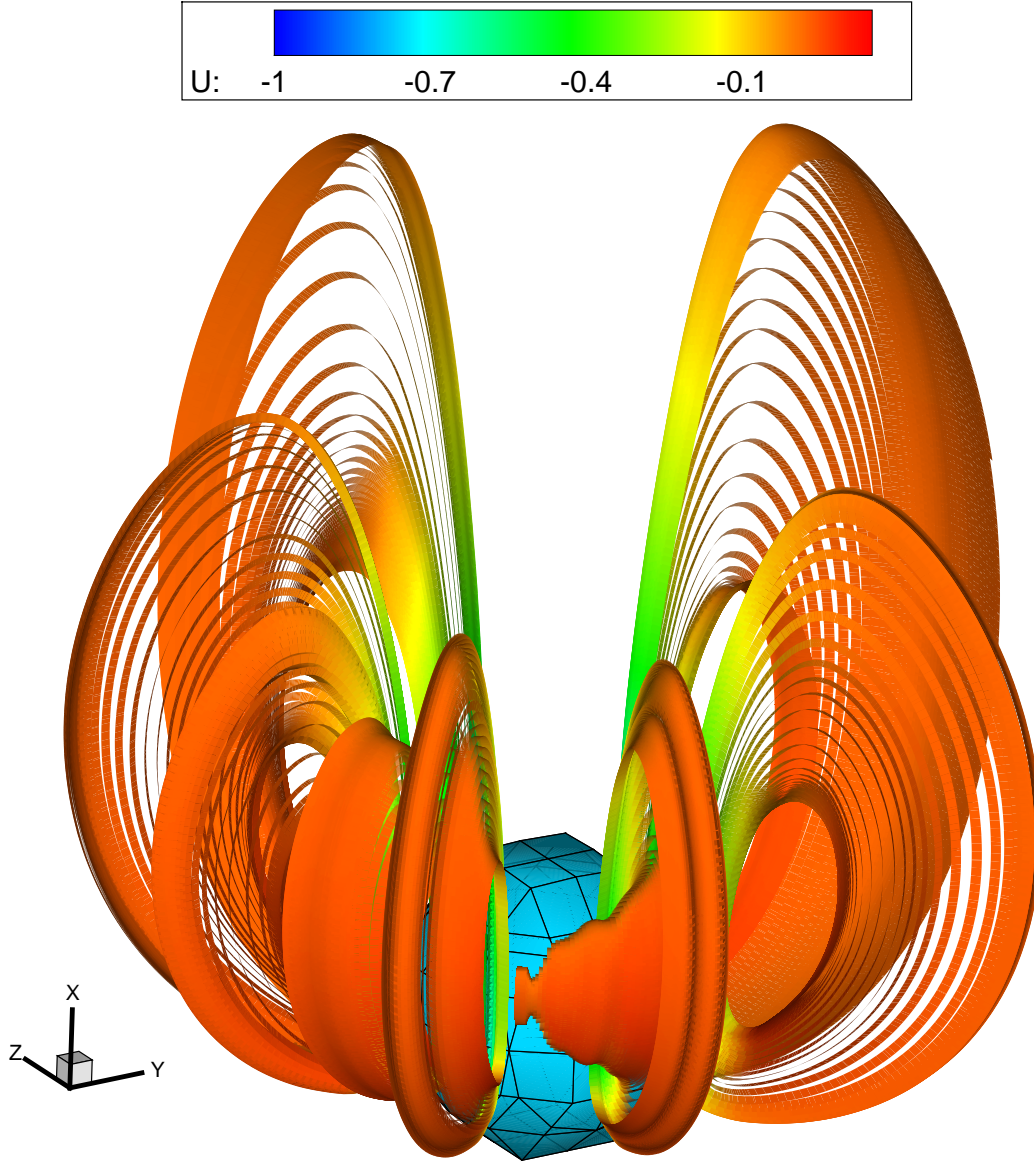


Figure 6.7: The streamlines colored by streamwise velocity of the falling sphere in a quiescent fluid at $t = 12$, obtained with the penalized Incompact3d solver, where $g = -20$, $\rho_b/\rho_f = 1.041$, $D = 1$, $(x, y, z) \in [0, 12] \times [0, 6] \times [0, 6]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 10^{-3}$ using AB2, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-2}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 10^{-2}$ and $Re \approx 100$.

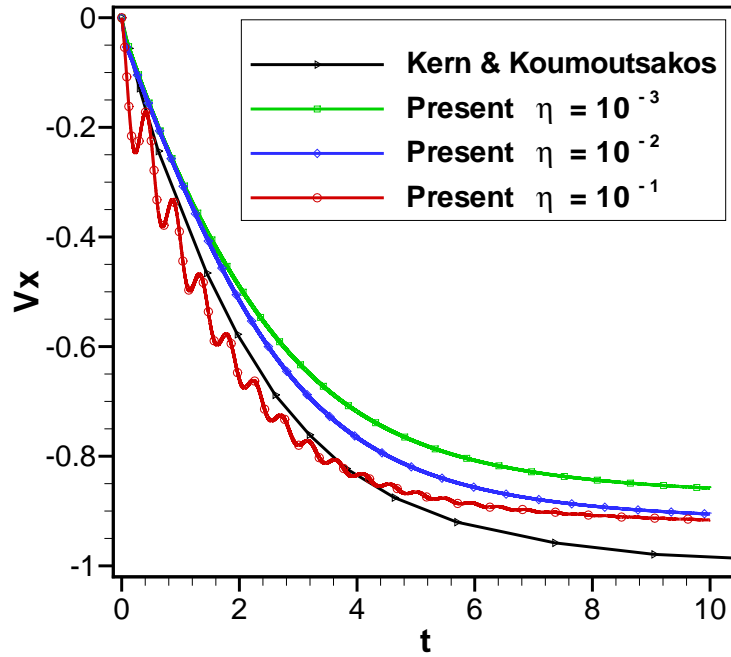


Figure 6.8: Comparison of the streamwise velocity V_x of the falling sphere ($g = 20$) using different penalization parameters η , with the reference simulation performed by Kern and Koumoutsakos [105] (black line). Colored lines represent the results of the present study performed with the penalized Incompact3d solver, with $\Delta t = 10^{-4}$ using AB3.

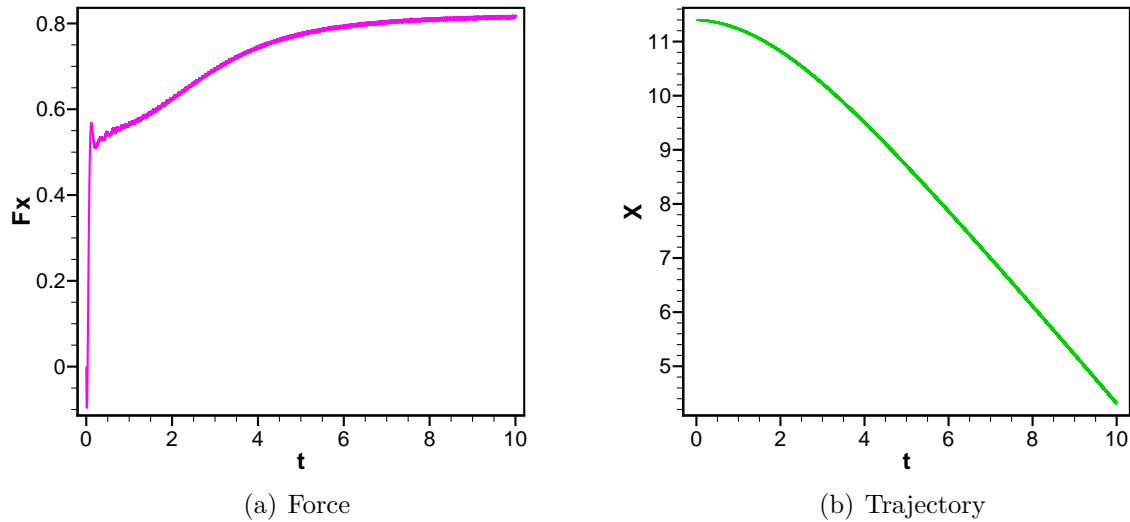


Figure 6.9: The corresponding streamwise force and the displacement of the falling sphere, represented in Fig. 6.8, by imposing the penalization parameter to $\eta = 10^{-2}$.

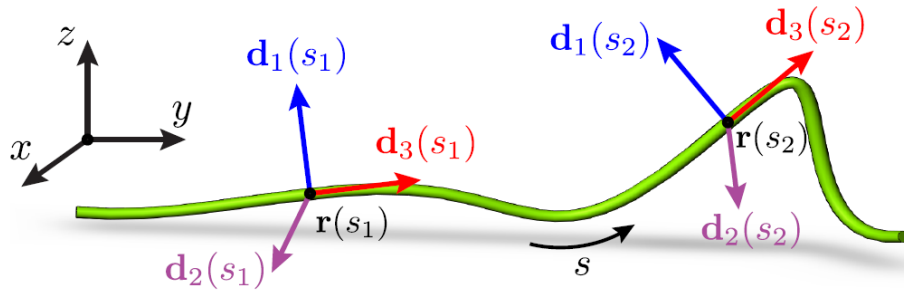


Figure 6.10: Backbone of the fish as a one-dimensional Cosserat medium. The inertial frame is denoted by (x, y, z) , the orthogonal body fitted coordinate (d_1, d_2, d_3) oriented along the body to be tangent in d_3 direction. Picture taken from Lazarus et al. [154].

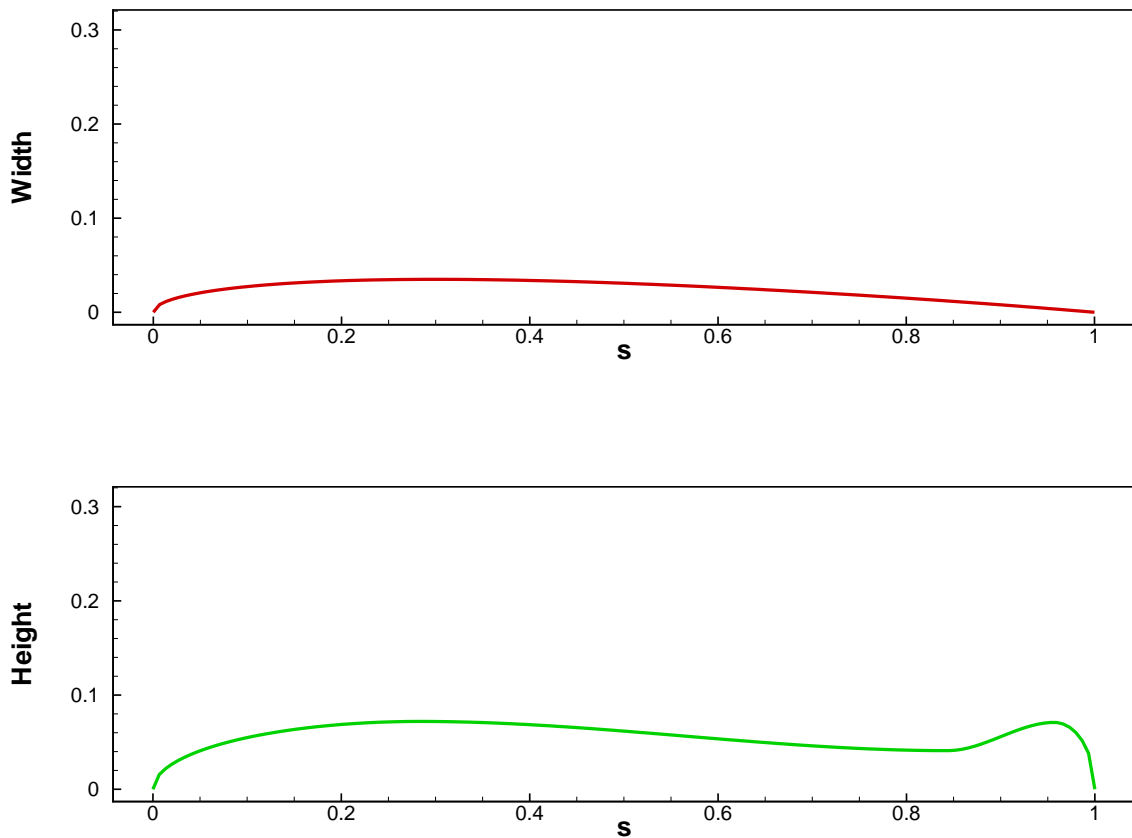


Figure 6.11: Profiles of the considered fish from top and side.

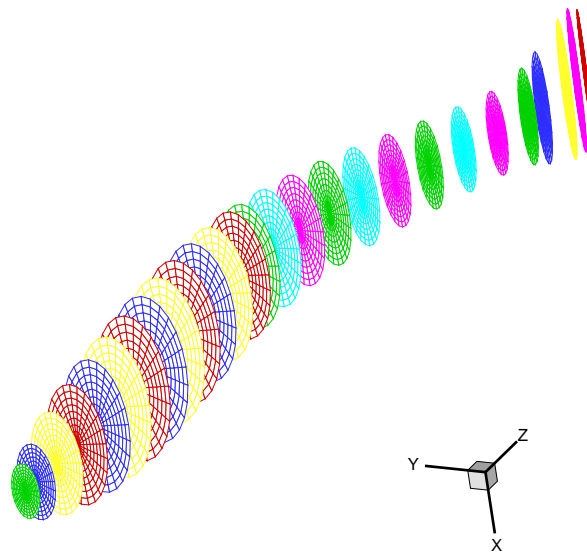


Figure 6.12: The fish is constructed by series of ellipses normal to the backbone of the considered fish. Each ellipse is covered by a structured grid.

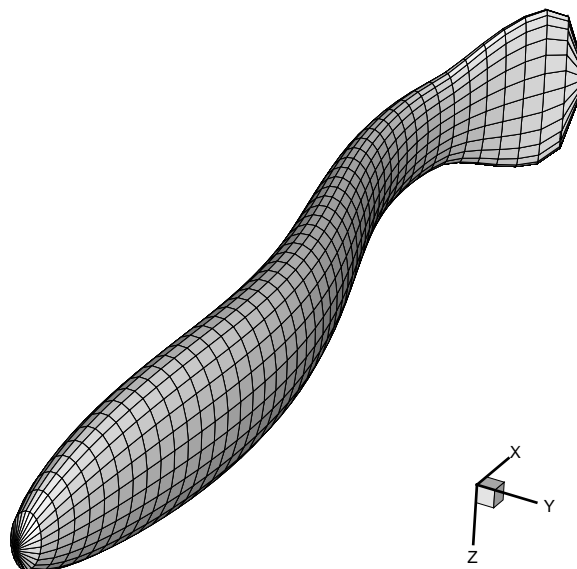


Figure 6.13: The surface of the considered fish is covered by a Lagrangian structured grid.

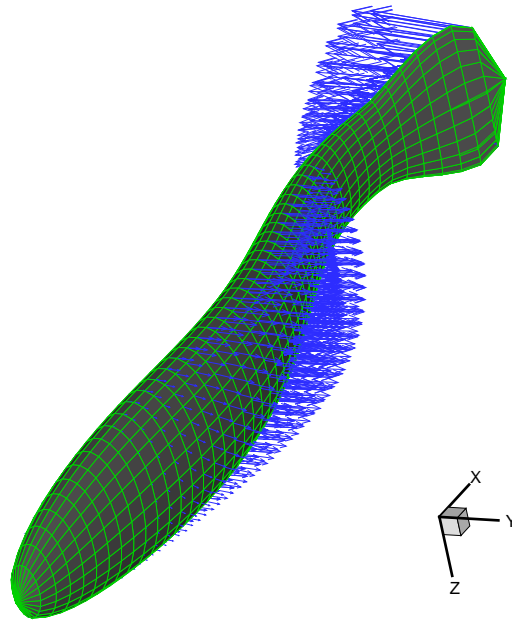


Figure 6.14: The corresponding velocities, evaluated by Eq. (6.52), of the swimming fish at the surface of the Lagrangian structured grid.

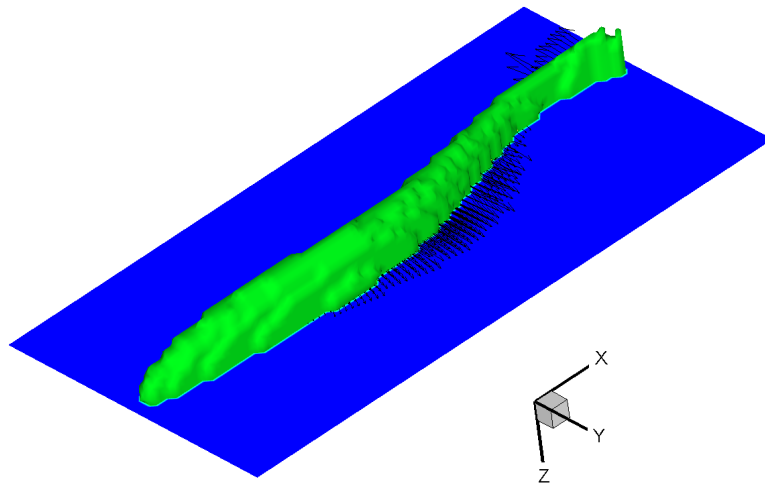
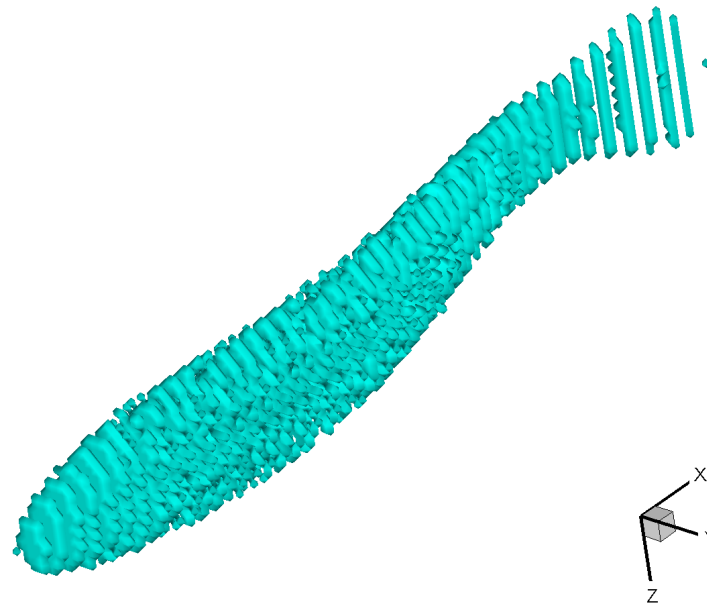
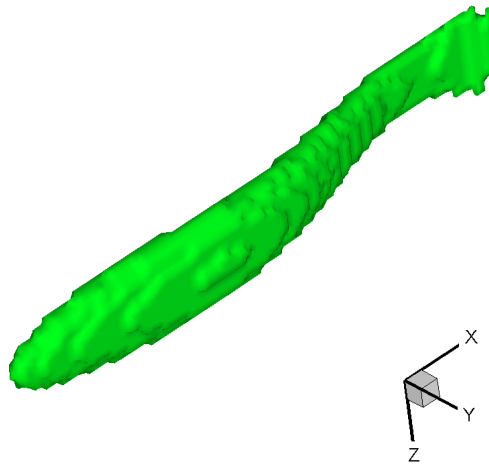


Figure 6.15: The interpolated mask function χ and the velocity components, on the Eulerian grid.



(a) Insufficient resolution



(b) Moderate resolution

Figure 6.16: The mask function with two different resolutions of the Lagrangian grid.

6.6 Conclusion

In this Chapter the three dimensional incompressible Navier–Stokes equations have been represented. Conservation of mass, momentum and energy, in discrete sense, in the inviscid limit are recalled. We have developed a second-order two-dimensional incompressible flow solver based on a projection method using half staggered grid for better understanding the difficulties of primitive variables for three dimensional extension. By using high order discretization and collocated grids, vorticity-stream function formulation proved to be more efficient than primitive variables in two dimensions. However, in three dimensions the primitive variables are more straightforward than the vorticity-velocity formulation. Therefore in three dimensional simulations we are using the Incompact3d open access code developed by Laizet and Lamballais [129]. The Incompact3d code is briefly reviewed then adapted to deal with fluid–structure interaction problems. The direct forcing is replaced by the implicit volume penalization method to take into account deformable bodies with imposed motion. A six degree of freedom simulator is also added to the solver. For the proposed penalized-projection method Eq. (6.35) is derived for evaluation of the pseudo pressure. Skew-symmetric discretization of the convective terms is used in the computations. Half staggered grid for pressure in conjunction with a high-order dissipative method is used to perform implicit large eddy simulations. Validation of the penalized-incompact3d is done by means of simulating a falling sphere. The results are in a satisfactory agreement with the reference simulation. Finally, some preliminary simulations of a three dimensional swimming fish are performed. The major bottlenecks in the computations are the resolution and the computation time. On the available machine just eight processors are optimal for the simulations. Future developments are adaptation of the code to available clusters to increase the number of CPUs in an efficient manner. The proposed law in Chapter 4 for rotation control of a two-dimensional swimmer must be extended to three dimensions. Another development is increasing the order of the immersed boundary method to be at least second order, because despite two dimensional simulations in three dimensions the resolution is limited. The evaluation of the hydrodynamic coefficients are crucial in three dimensional simulations. The presented control volume law in Chapter 3, for evaluation of the hydrodynamic coefficients must also be extended to three dimensions to examine if it can perform better.

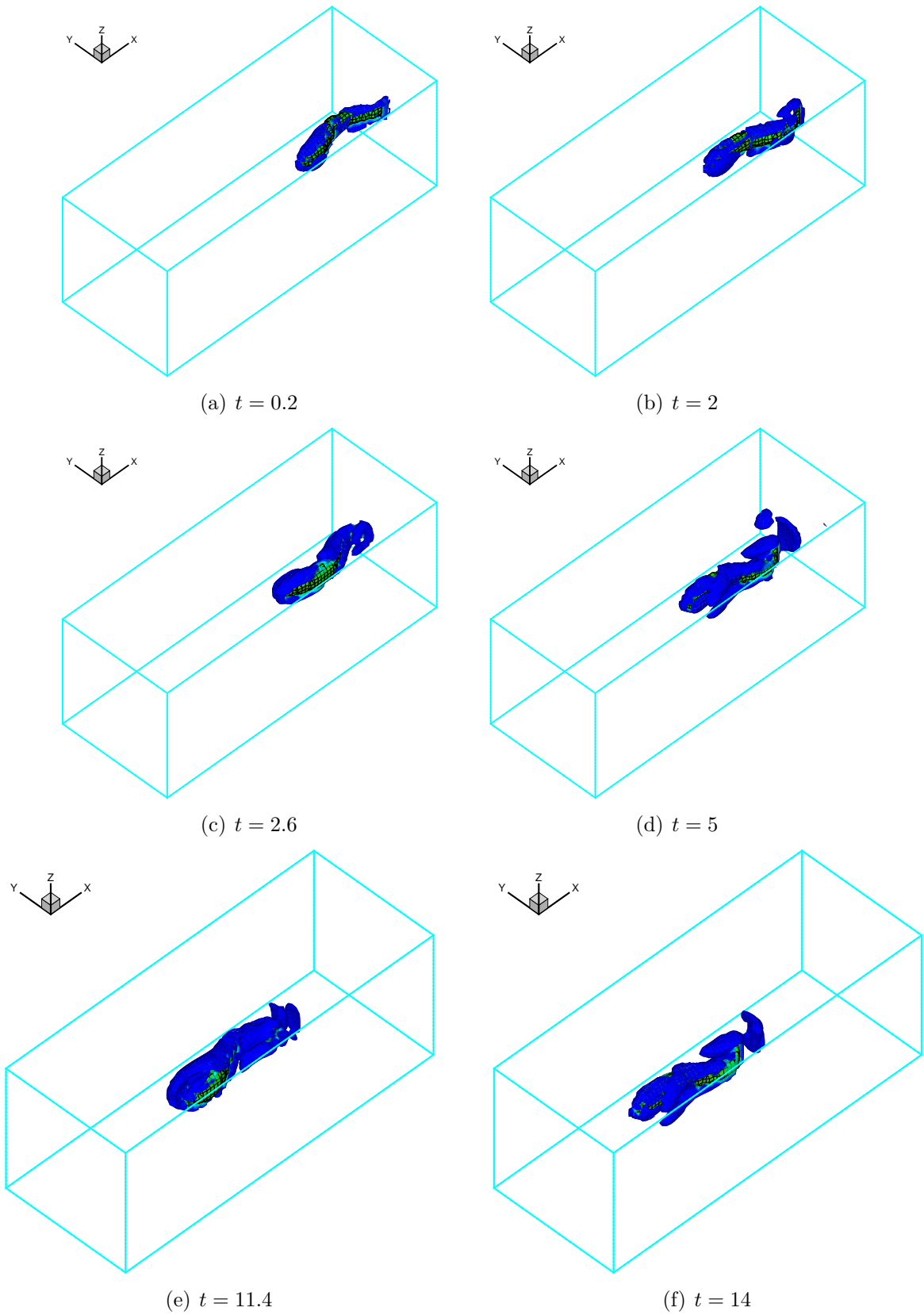


Figure 6.17: Q iso-surfaces of the swimming fish obtained with the penalized Incompact3d solver, where $l = \lambda = f = 1$, $\rho_b = \rho_f$, $(x, y, z) \in [0, 3] \times [0, 1] \times [0, 1]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 2 \times 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-3}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 2 \times 10^{-3}$ and $Re \approx 100$.

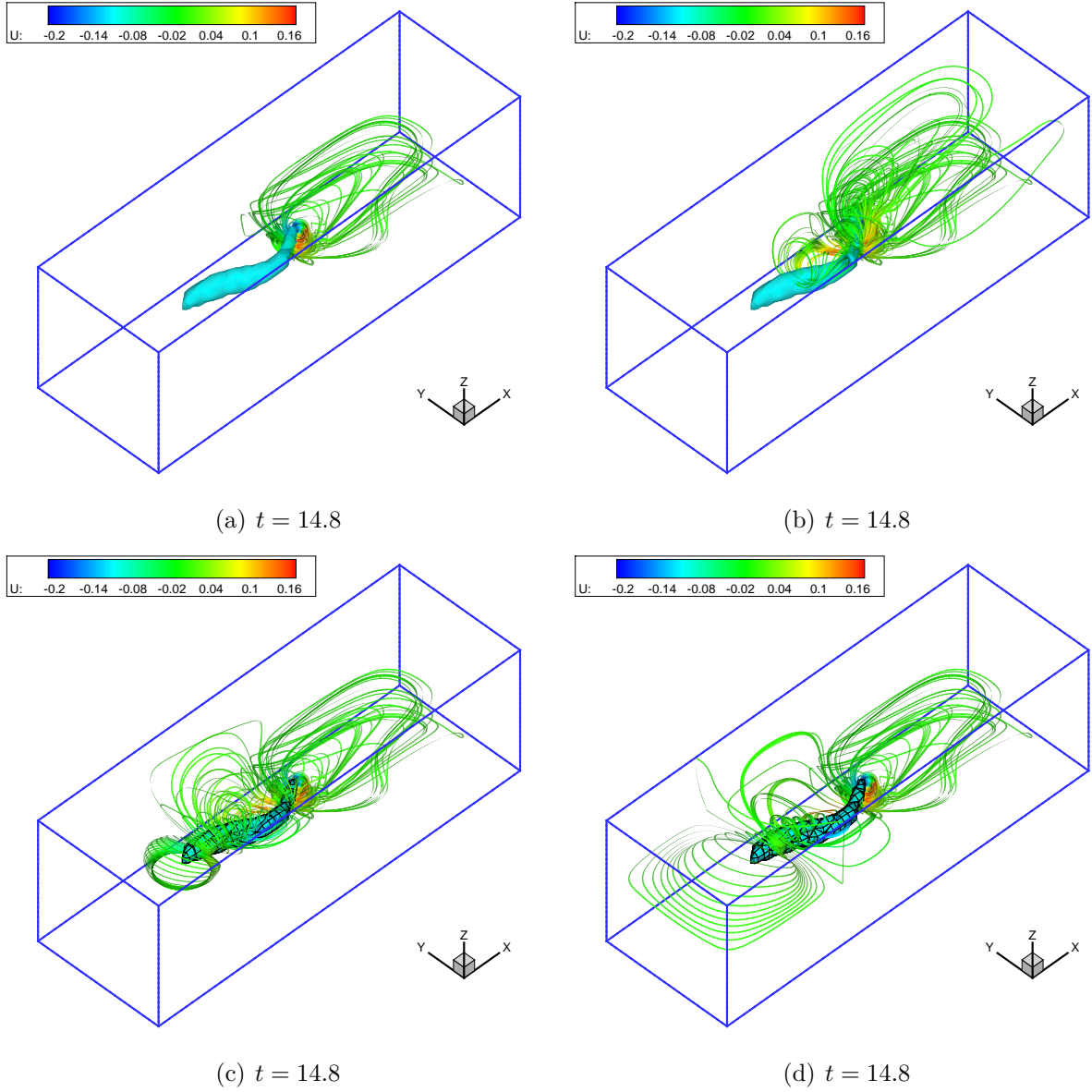


Figure 6.18: The streamlines colored by streamwise velocity of the swimming fish obtained with the penalized Incompact3d solver, where $l = \lambda = f = 1$, $\rho_b = \rho_f$, $(x, y, z) \in [0, 3] \times [0, 1] \times [0, 1]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 2 \times 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-3}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 2 \times 10^{-3}$ and $Re \approx 100$.

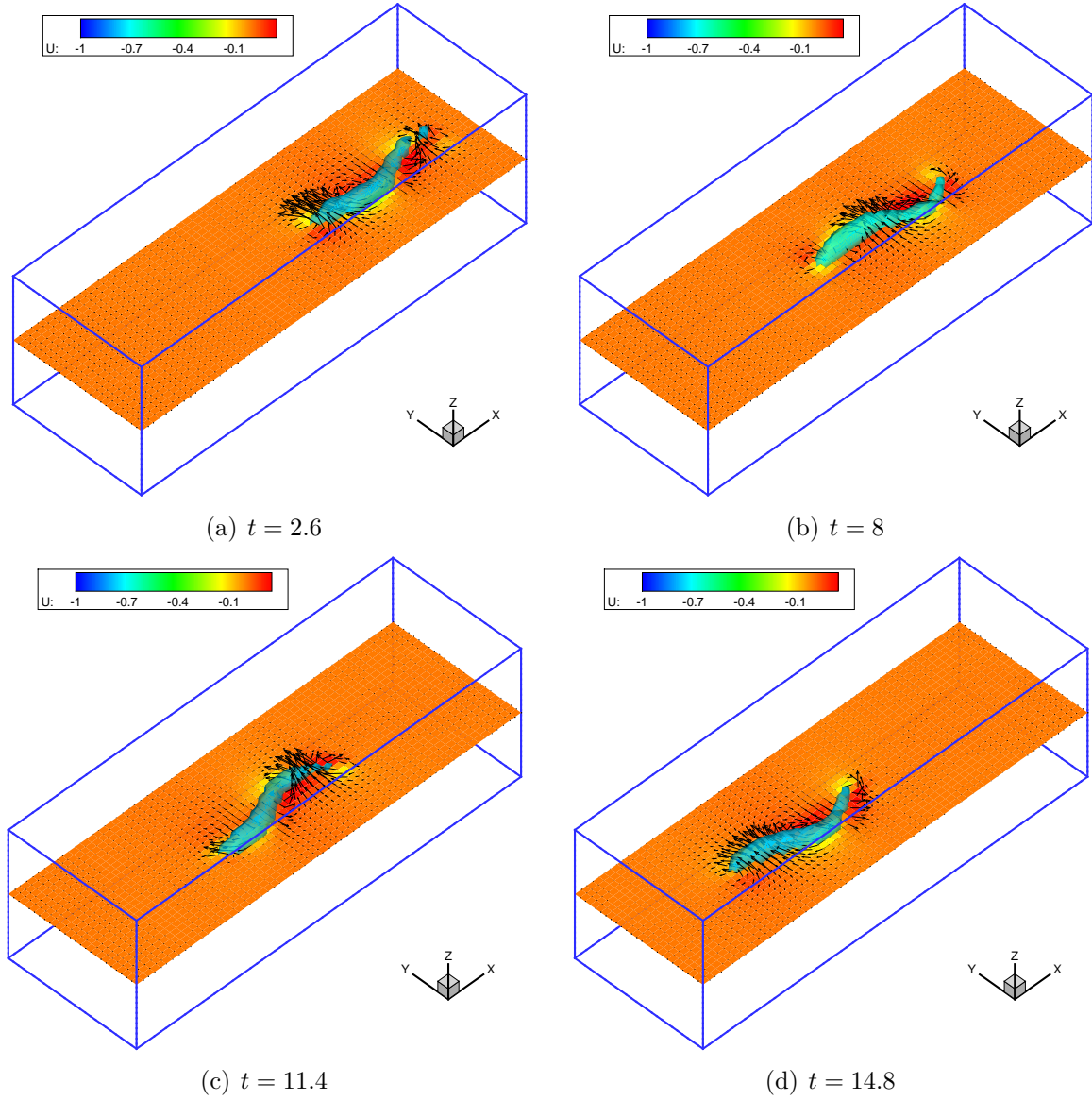


Figure 6.19: The z -mid velocity field of the swimming fish obtained with the penalized Incompact3d solver, where $l = \lambda = f = 1$, $\rho_b = \rho_f$, $(x, y, z) \in [0, 3] \times [0, 1] \times [0, 1]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 2 \times 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-3}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 2 \times 10^{-3}$ and $Re \approx 100$.

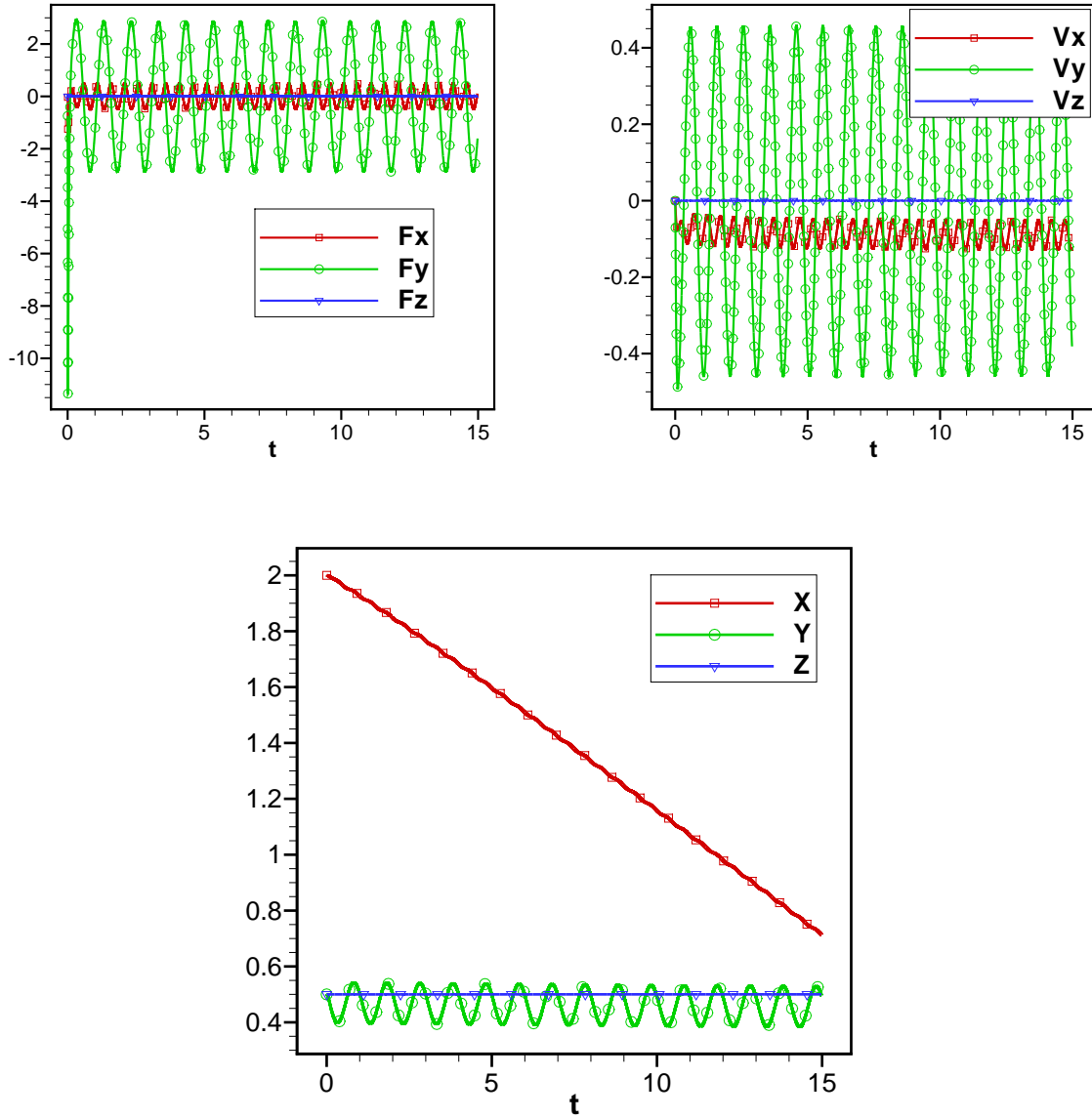


Figure 6.20: The forces (top-left), velocities (top-right) and the trajectories (bottom) of the swimming fish obtained with the penalized Incompact3d solver, where $l = \lambda = f = 1$, $\rho_b = \rho_f$, $(x, y, z) \in [0, 3] \times [0, 1] \times [0, 1]$, BC (2-1-1) is imposed (see Table 6.2), $\Delta t = 2 \times 10^{-4}$ using AB3, resolution $257 \times 101 \times 101$, penalization parameter $\eta = 10^{-3}$, $\delta_{\text{filter}} = 10^{-3}$, $\nu = 2 \times 10^{-3}$ and $Re \approx 100$. The reference point is the head.

Chapter 7

Conclusion and perspectives

“It must be admitted that the principal result of fifty years of turbulence research is the recognition of the profound difficulties of the subject [96].”

S. A. Orszag (1970)

The subject of present investigation is the simulation of forced deformable bodies interaction with an incompressible flow. As an application quantification of a swimming fish is considered. To this end an efficient numerical algorithm have been proposed. The incompressible Navier-Stokes equation are considered as the mathematical model. For two-dimensional simulations the vorticity-stream function formulation proved to be more efficient. Explicit fourth-order Rung–Kutta method is used for time integration of the governing equations. To achieve high accuracy, compact finite differences are applied to the spatial terms. By using a uniform Cartesian grid a new fourth-order direct solver was presented for the solution of the Poisson equation, which combines finite differences with FFT in alternative directions. In order to introduce a deformable body in fluid flow, the volume penalization method is applied to the solution of the Navier–Stokes equations as a forcing term. Even if the penalization method is shown to have between first and second order accuracy in space, an important advantage of this method is that the evaluation of the hydrodynamic coefficients is straightforward. However, proper denoising of the hydrodynamic coefficients is crucial in dealing with fluid/solid interaction problems via the volume penalization method. Another advantage of the volume penalization method is the possibility of flow simulation around almost arbitrary geometries with an imposed motion by using uniform Cartesian grids. For examination of the error due to the penalization term, the Taylor–Couette flow was considered and between first and second order accuracy in space was proved. Then for validation of the fluid–structure interaction, falling of a cylinder and an ellipse in a quiescent fluid (due to terrestrial gravity) was studied. Simu-

lation of the fish in forward gait is next considered for further validation of the proposed algorithm. A Lagrangian structured grid covers exactly the fish body, which is interacting with the surrounding fluid due to the hydrodynamic forces and the torque calculated on the Eulerian reference grid. Good agreement is observed with the results reported by Gazzola et al. [144]. An efficient law for curvature control of an anguilliform fish, swimming toward a predefined goal, is proposed which is based on geometrically exact theory of nonlinear beams. By using the quaternions for rotation description, the exact theory of nonlinear beams is proved to be accurate, efficient and straightforward. With the proposed rotation control law, the motionless fish executes a sharp 180° turn within an area of about 1.3 times its body length. Validation of the developed method shows the efficiency and expected accuracy (between first and second) of the algorithm for rotation control of an anguilliform swimmer and also for a variety of fluid/solid interaction problems. A perspective for future works is the enhancement of the rotation control law. The FORTRAN code is open access [171], the interested users are first encouraged to try the second-order solver over the finest possible grid, then investigate the effect of increasing the order from second to fourth on the same or a coarser grid. However increasing the accuracy order of the immersed boundary method is a challenging task. For high-order IBMs implemented to finite difference solvers we refer to Linnick and Fasel [98], Seo and Mittal [147] and Bonfigli [148].

In the Chapter 5, multiresolution analysis is applied to the algorithm to deal with two-dimensional flows interacting with deformable bodies, on adaptive grids. This method restricts the computational effort to the regions where high gradients of the flow variables are present. The method is based on Harten's [64] point value analysis which allows to reduce the number of active grid points significantly by refining/coarsening the grid automatically. This can be done through nonlinear thresholding of the wavelet coefficients in a one-to-one correspondence with the grid points. A second-order central finite difference method with symmetric stencil on an adaptive Cartesian grid is used for spatial discretization of the equations. For validation of the adaptive solver, simulation of dipole wall collision is performed by both uniform and adaptive solvers, the results are in good agreement. After validation of the proposed algorithm an extension to deal with fluid interaction with forced deformable bodies, i.e., swimming of a fish, is done by implementing the volume penalization method. The results of fish swimming in forward gait are compared with those of Gazzola et al. [144] where a uniform grid is used. The obtained results show that the CPU-time of the adaptive simulations can be significantly reduced with respect to simulations on a uniform grids. Nevertheless the accuracy order of the underlying numerical scheme is preserved. Implementation of a tree data-structure or a hash table for memory deallocation is proposed as a perspective for further studies, we refer to the work of Hejzialhosseini et al. [136]. Extending the code to include a dynamic and distributed memory for parallel computations with message passing interface (MPI) is another subject for future investigations.

For three dimensional simulations the primitive variables are more straightforward than the vorticity-velocity formulation. Therefore in Chapter 6 a second-order two-dimensional incompressible flow solver (based on a projection method) is developed for better understanding the difficulties of primitive variables and grid staggering. After that for three dimensional simulations the Incompact3d open access code is used, which is originally developed by Laizet and Lamballais [129]. Half staggered grid for pressure in conjunction with a high-order dissipative method are used to perform implicit large eddy simulations. Then in the present investigation, the Incompact3d code is adapted to deal with fluid–structure interaction problems. The direct forcing is replaced by the implicit volume penalization method to take into account deformable bodies (with an imposed motion) interaction with a fluid. A six degree of freedom simulator is also added to the solver. For evaluation of the pseudo pressure in the proposed penalized-projection method, a Poisson equation (6.35) is derived. Validation of the penalized-incompact3d solver is done by simulating a falling sphere. Finally, some preliminary simulations of a three dimensional swimming fish are performed. The major bottlenecks in the computations are the resolution and the computation time. A future development is adaptation of the code to available clusters, to use more computational cores in an efficient manner and then increasing the Reynolds number. The proposed law in Chapter 4, for rotation control of a two-dimensional swimmer, must be extended to three dimensions. Another development is increasing the order of the immersed boundary method to be at least second order, because in contrast to two dimensional simulations in three dimensions the resolution is much more limited, we refer to Linnick and Fasel [98], Seo and Mittal [147] and Bonfigli [148] for high-order IBMs. The evaluation of the hydrodynamic coefficients is crucial in the simulations. The proposed control volume law for evaluating the hydrodynamic coefficients in Chapter 3 may also be extended to three dimensions to examine its performance. The codes are developed in FORTRAN and are open access [171]-[172].

Chapter 8

Résumé de thèse en français

“ En Mars 1922, lors d’un dîner Paul Valéry posa la question à Albert Einstein, Avez-vous un petit carnet où vous notez vos idées? Einstein, avec son espièglerie habituelle mâtinée d’une profonde sagesse, lui répondit qu’il n’a pas besoin de carnet, car « Oh ! Vous savez, une idée, c’est si rare ! ».”

Marie Farge [152] (2011)

Dans cette étude, une méthode numérique précise et efficace est proposée pour la simulation de corps déformables interagissant avec un écoulement incompressible. L’application principale de cet algorithme concerne la simulation numérique de la nage d’un poisson. La quantification de l’écoulement autour des animaux aquatiques est une difficulté, quant à sa modélisation et à sa simulation, dans le domaine de la mécanique des fluides numérique. Avec plus de 32000 espèces différentes, un des problèmes est la représentation de la cinématique du poisson. La simulation de l’écoulement incompressible est aussi une problématique où l’efficacité du solveur de Poisson devient crucial. Le troisième problème est le couplage entre le fluide et le corps déformable et mobile. Un poisson nage en exerçant des forces et des moments dans l’eau qui l’environne, en s’opposant à la résistance hydrodynamique, i.e., la traînée. Dans certains cas, comme la nage anguilliforme, un mouvement ondulatoire du corps se développe de la tête vers la queue. L’amplitude de cette oscillation augmente au fur et à mesure qu’elle se propage vers la queue. La majorité des poissons nage grâce à une ondulation/oscillation de leur corps et de leurs nageoires (aillette/aileron). Dans l’étude présente, pour répondre aux problèmes posés, les équations de Navier–Stokes, sont considérées dans leur formulation vorticit -fonction de courant. Ensuite, elles sont discr tis es temporellement et spatialement   l’aide respectivement d’un sch ma d’ordre quatre de Runge–Kutta et par des diff rences finies compactes d’ordre quatre. Conjointement   l’utilisation d’un maillage uniforme, nous proposons un nouveau solveur direct au quatri me

ordre pour l'équation de Poisson, permettant de garantir la contrainte d'incompressibilité au niveau du zéro machine sur une grille à pas d'espace optimale. L'introduction d'un corps déformable et mobile dans l'écoulement de fluide est réalisée au moyen d'une méthode de pénalisation de volume. La déformation du corps est imposée par l'utilisation d'un maillage lagrangien structuré et mobile qui interagit avec le fluide environnant en raison des forces hydrodynamiques et du moment (calculés sur le maillage eulérien de référence). Une loi efficace de contrôle de la courbure pour un poisson anguilliforme nageant vers un objectif prescrit est proposée. La loi de contrôle est utilisée pour changer la direction de le nage du poisson et elle est basée sur la théorie exacte des poutres non-linéaires.

8.1 Modèle mathématique

Les équations de Navier–Stokes gouvernent les écoulements incompressibles et newtoniens. En utilisant les variables primitives, elles sont composées des équations de quantité de mouvement :

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho_f} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F} \quad , \quad \mathbf{x} \in \Omega \in \mathbb{R}^3 \quad (8.1)$$

et de continuité :

$$\nabla \cdot \mathbf{u} = 0 \quad (8.2)$$

Pour les écoulements bidimensionnels, les équations de Navier–Stokes (8.1) - (8.2) sont considérées dans leur formulation vorticité et fonction de courant :

$$\partial_t \omega + (\mathbf{u} \cdot \nabla) \omega = \nu \nabla^2 \omega + \nabla \times \mathbf{F} \quad , \quad \mathbf{x} \in \Omega \in \mathbb{R}^2 \quad (8.3)$$

où $\omega(\mathbf{x}, t) = \nabla \times \mathbf{u} = v_x - u_y$ est la vorticité. Le champ de vitesse est donné par

$$\mathbf{u} = (u, v) = (\partial_y \psi, -\partial_x \psi)$$

où ψ est la fonction de courant, satisfaisant une équation de Poisson :

$$-\nabla^2 \psi = \omega \quad (8.4)$$

qui garantit la contrainte d'incompressibilité (8.2). Dans cette étude, l'introduction d'un corps déformable et mobile dans l'écoulement de fluide est réalisée au moyen d'une *méthode de pénalisation de volume* proposée par Arquis (1984) [40], Angot et al. (1999) [73] puis Khadra et al. (2000) [75]. Le terme de pénalisation sur la vitesse est défini par :

$$\mathbf{F} = -\eta^{-1} \chi(\mathbf{u} - \mathbf{u}_B) \quad (8.5)$$

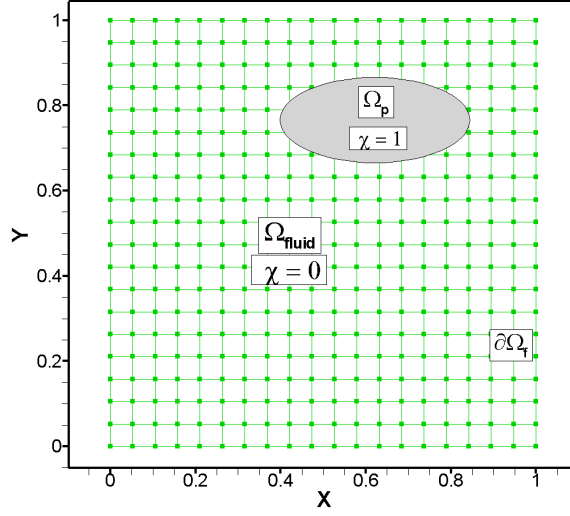


Figure 8.1: Domaine de la solution et du corps immergé, $\Omega = \Omega_f \cup \Omega_p$.

où $\mathbf{u}_B(\mathbf{x}, t)$ est le champ de vitesse imposé dans le corps déformable, χ est la fonction caractéristique de l'objet et η est le coefficient de perméabilité. La fonction caractéristique χ est définie comme :

$$\chi(\mathbf{x}, t) = \begin{cases} 1 & \mathbf{x} \in \Omega_p \\ 0 & \mathbf{x} \in \Omega_f \end{cases} \quad (8.6)$$

où Ω_f représente le domaine fluide et Ω_p représente le corps solide ou déformable immergé dans le fluide. La fonction caractéristique χ est sans dimension et décrit le corps immergé dans le fluide, comme illustré sur la Figure 8.1. Selon Carbou et Fabrie [92] quand $\eta \rightarrow 0$, la solution numérique pénalisée \mathbf{u} converge vers la solution exacte du problème pénalisé $\mathbf{u}_{\text{exact}}$, avec un taux de convergence de l'ordre $O(\sqrt{\eta})$.

Notons par ailleurs que Pasquetti et al. (2008) [121], Minguez (2008) [122], Kolomenskiy et Schneider (2009) [126] et quelques autres chercheurs proposent d'utiliser une fonction caractéristique filtrée $\bar{\chi}$, pour stabiliser le schéma de discrétisation, qui est indispensable en utilisant des schémas d'ordre élevé. Pasquetti et al. (2008) [121] rappellent que le filtrage de type “*raised cosine*” utilisé dans la thèse de Forestier (2000) [79] (communément utilisé dans l'espace spectral) s'écrit dans l'espace physique comme :

$$\bar{\chi}_{i,j} = (2\chi_{i,j} + \chi_{i+1,j} + \chi_{i-1,j} + \chi_{i,j+1} + \chi_{i,j-1})/6 \quad (8.7)$$

Ce filtre est aussi utilisé par Shuman (1957) [15] en météorologie.

8.2 Dynamique d'un objet mobile

Outre la possibilité de modéliser des géométries complexes sur des maillages cartésiens à moindre coût, la méthode de pénalisation a l'avantage de donner facilement accès aux forces et moments hydrodynamiques qui s'exercent sur le corps, sans intégration du tenseur des contraintes σ sur la surface du corps. Angot et al. (1999) [73] montrent que les forces sont données par:

$$\mathbf{F} = \oint_{\partial\Omega_s} \sigma \cdot \mathbf{n} \, dl = \lim_{\eta \rightarrow 0} \frac{\rho_f}{\eta} \int_{\Omega_s} \chi(\mathbf{u} - \mathbf{u}_B) \, ds + \rho_f S_{pen} \ddot{\mathbf{X}}_{ref} \quad (8.8)$$

De façon similaire le moment $[N.m]$ appliqué est donné par :

$$M_{ref} = \oint_{\partial\Omega_s} \mathbf{r} \times \sigma \cdot \mathbf{n} \, dl = \lim_{\eta \rightarrow 0} \frac{\rho_f}{\eta} \int_{\Omega_s} \chi \mathbf{r} \times (\mathbf{u} - \mathbf{u}_B) \, ds + \frac{\rho_f}{\rho_s} I_{zz} \ddot{\theta}_{ref} \quad (8.9)$$

où $I_{zz} = \int \mathbf{r}^2 dm$ est le moment d'inertie du corps autour d'un point de référence, \mathbf{n} est le vecteur unitaire dirigé vers l'extérieur du corps normal à $\partial\Omega_s$, θ est l'angle de rotation par rapport au point de référence, les points représentent la dérivée seconde temporelle et S_{pen} est la surface de la zone pénalisée.

Dans le cas des problèmes d'interaction fluide-structure, l'utilisation de la méthode de pénalisation de volume avec une résolution modérée et les calculs numériques des coefficients hydrodynamiques suivant des approximations (8.8) et (8.9), entraîne des oscillations des coefficients hydrodynamiques au cours des processus itératifs qui perturbent la convergence du calcul. Les oscillations produisent des bruits numériques qui peuvent provoquer la divergence de la solution ou la convergence vers une solution imprécise. Une méthode efficace pour éliminer ces bruits consiste à appliquer un filtre passe-bas du type *lissage exponentiel d'ordre deux* introduit par Holt (1957) [14], qui est utilisé régulièrement pour filtrer les données temporelles :

$$\hat{F}^n = \alpha F^n + (1 - \alpha)(\hat{F}^{n-1} + b^{n-1}) \quad , \quad n = 3, 4, \dots \quad (8.10)$$

$$b^n = \beta(\hat{F}^n - \hat{F}^{n-1}) + (1 - \beta)b^{n-1} \quad , \quad (\alpha, \beta) \in [0, 1] \quad (8.11)$$

où $\hat{F}^1 = F^1$, pour $n = 2$ on peut utiliser les équations (8.10) et (8.11) avec $\alpha = \beta = 1$. Par la suite $\alpha = 1 - (1 - \delta)^2$ et $\beta = \delta^2/\alpha$ peuvent être utilisés où le paramètre de filtrage δ représente une gamme étroite ($\delta \ll 1$). Selon notre expérience $\delta \in [10^{-4}, 10^{-2}]$ peut être utilisé pour le filtrage des coefficients hydrodynamiques. Les valeurs plus faibles ajoutent un effet d'amortissement relativement fort sur le mouvement de corps. Néanmoins, en utilisant des valeurs plus grandes il y a des risques de divergence dans les simulations. Une analyse de sensibilité est alors nécessaire pour obtenir des résultats fiables.

La dynamique d'un corps immergé dans un fluide est gouvernée par la loi de Newton:

$$\Sigma(\mathbf{F}_H + \mathbf{F}_G) = m\ddot{\mathbf{X}}_{ref} \quad (8.12)$$

où les forces appliquées sont décomposées en deux composantes, i.e., les forces hydrodynamiques \mathbf{F}_H et les forces dues à la gravité $\mathbf{F}_G = S_{pen}(\rho_b - \rho_f)\mathbf{g}$. La loi de Newton peut directement être intégrée, pour donner les positions du centre de gravité (cg) au cours du temps. Si l'on suppose que \mathbf{F} est constant durant un pas de temps discret (t^n, t^{n+1}) , on obtient alors :

$$\Delta\mathbf{X}_{ref} = \frac{1}{2} \frac{\mathbf{F}^n}{m} \Delta t^2 + \mathbf{V}^n \Delta t \quad (8.13)$$

et $V^{n+1} = V^n + \ddot{X} \Delta t$. Le mouvement de rotation d'un objet est décrit par loi d'Euler :

$$\Sigma M_{ref} = \frac{d}{dt}(J_{ref}\dot{\theta}) \quad (8.14)$$

où M_{ref} est le moment appliqué autour d'un point de référence. Si le point de référence choisi n'est pas identique avec le centre de gravité (cg), le moment dû à la force de gravité (flottabilité) doit être rajouté à ΣM_{ref} dans l'équation (8.14). En présence des forces volumiques, le choix du centre de gravité (cg) comme point de référence permet de simplifier les calculs du moment, i.e., seulement le moment dû aux forces hydrodynamiques \mathbf{F}_H doit être intégré autour du point de référence.

Dans cette étude, le centre de gravité (cg) est utilisé comme point de référence pour les simulations de sédimentation des objets considérés, comme la chute d'une ellipse ou d'un cylindre dans un fluide. Cependant, pour les simulations de nage de poisson ($\rho_b = \rho_f$) la flottabilité est égale à zéro et ne joue aucun rôle. C'est pourquoi le choix de la tête du poisson comme point de référence peut simplifier l'intégration des équations (8.28), (8.30) et (8.32), pour obtenir la cinématique de la colonne vertébrale, sans avoir besoin du moment dû à la flottabilité. L'intégration temporelle de l'équation (8.14) sans prendre en compte les variations de moment d'inertie et M_{ref} , donne le nouvel angle du corps considéré par rapport au point de référence :

$$\Delta\theta = \frac{1}{2} \ddot{\theta}^n \Delta t^2 + \dot{\theta}^n \Delta t \quad (8.15)$$

où $\ddot{\theta} = M/J$ et $\dot{\theta}^{n+1} = \dot{\theta}^n + \ddot{\theta} \Delta t$ (les points représentent les dérivées temporelles). Les équations (8.13) et (8.15) décrivent un mouvement avec trois degrés de liberté pour le corps considéré. Les équations (8.8) et (8.9) fournissent les forces et le moment exercés pour l'intégration du système des équations aux dérivées partielles (EDP) formé par les équations (8.12) et (8.14). Le filtrage des coefficients hydrodynamiques doit être fait selon l'équation (8.10). Des conditions initiales convenables sont nécessaires. Dans cette étude, on utilise un schéma d'ordre un $O(1)$ pour l'intégration temporelle des équations de la

dynamique. La même méthode d'intégration est utilisée par Kolomenskiy et Schneider (2009) [126] et Gazzola et al. (2011) [144] pour la dynamique d'un corps mobile où la pénalisation de volume est également employée.

8.3 Discrétisation temporelle et spatiale

L'équation (8.3) est discrétisée temporellement à l'aide d'un schéma classique d'ordre quatre de type *Runge-Kutta* (RK4). Pour la discrétisation spatiale, des schémas aux *différences finies compactes* d'ordre deux et quatre, présentés par Lele (1992) [55], sont utilisés. La méthode compacte d'ordre quatre est appliquée avec succès aux équations de Navier-Stokes (8.1) avec Abide et Viazzo [97] en utilisant les variables primitives. Sur une maille uniforme avec un pas d'espace égal à h , la dérivée première avec différents ordres de précision peut se construire avec :

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = a \frac{f_{i+1} - f_{i-1}}{2h} + b \frac{f_{i+2} - f_{i-2}}{4h} + c \frac{f_{i+3} - f_{i-3}}{6h} \quad (8.16)$$

et de façon similaire pour la dérivée seconde on a :

$$\begin{aligned} \beta f''_{i-2} + \alpha f''_{i-1} + f''_i + \alpha f''_{i+1} + \beta f''_{i+2} = \\ a \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + b \frac{f_{i+2} - 2f_i + f_{i-2}}{4h^2} + c \frac{f_{i+3} - 2f_i + f_{i-3}}{9h^2} \end{aligned} \quad (8.17)$$

on se réfèrera à l'annexe A pour les coefficients et leurs traitements près des bords. En choisissant $\alpha = \beta = 0$ dans les équations (8.16) et (8.17) des schémas explicites en résultent. Avec $\beta \neq 0$ ou $\alpha \neq 0$ les schémas sont implicites et un système linéaire d'équations de la forme suivante est obtenu :

$$[A]\mathbf{f}' = \mathbf{R}$$

Avec $\beta \neq 0$ et $\alpha \neq 0$, le système linéaire d'équations possède cinq diagonales (pentadiagonal) pour la matrice des coefficients $[A]$. Avec $\beta = 0$ le système obtenu comporte trois diagonales (tridiagonal) pour la matrice des coefficients $[A]$. Les systèmes d'équations linéaires à trois ou cinq diagonales, peuvent être résolus avec des méthodes efficaces. Deux algorithmes directs de décomposition de matrice du type "*inférieur-supérieur*" (Lower-Upper) sont présentés dans l'annexe F.

8.4 Solveur de Poisson rapide

Du fait de l'utilisation d'un maillage uniforme (à pas d'espace identique), nous proposons un nouveau solveur direct au quatrième ordre pour l'équation de Poisson, permettant de maintenir la contrainte d'incompressibilité au niveau du zéro machine sur une grille à

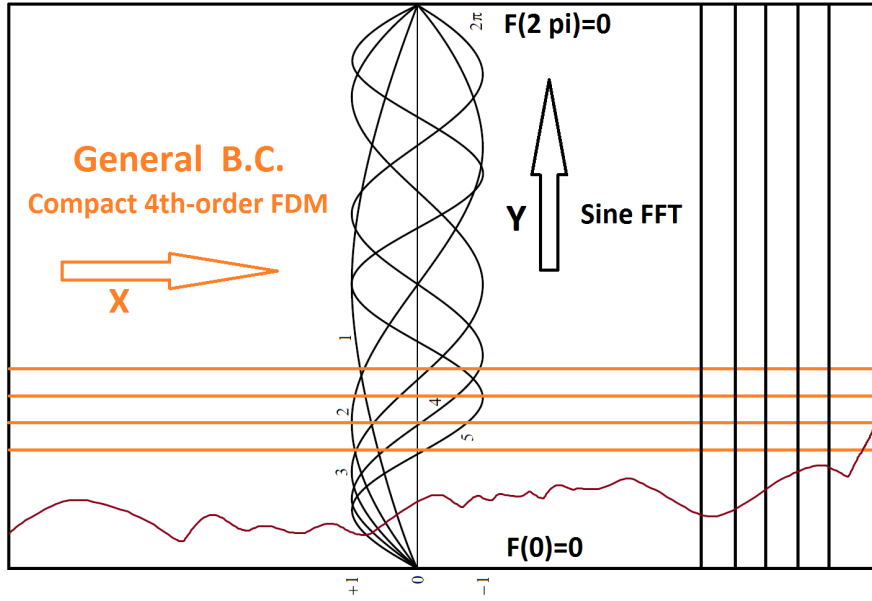


Figure 8.2: Une représentation schématique du domaine de la solution pour le solveur rapide de l'équation de Poisson. Les opérations dans les directions x et y sont découplées. Dans la direction x des conditions aux limites générales peuvent être utilisées grâce aux schémas aux différences finies. Dans la direction y des conditions aux limites d'imperméabilité et de glissement (Dirichlet homogène, i.e., $\psi = \omega = 0$) sont imposées permettant d'utiliser la transformée en *sinus*.

pas d'espace optimale. Pour déduire un schéma compact d'ordre quatre, pour l'équation de Poisson $-\nabla^2\psi = \omega$, sur un maillage uniforme de dimension $N_x \times N_y$, on utilise l'approximation suivante :

$$\frac{\partial^2\psi}{\partial x^2} = \delta_x^2\psi - \frac{\Delta x^2}{12} \frac{\partial^4\psi}{\partial x^4} + O(\Delta x^4) \quad (8.18)$$

pour la direction x , où δ_x^2 représente une approximation centrée d'ordre deux pour la dérivée seconde. En remplaçant (8.18) dans l'équation de Poisson, on obtient :

$$\left(\delta_x^2 - \frac{\Delta x^2}{12} \frac{\partial^4}{\partial x^4} + \partial_{yy}\right)\psi = -\omega \quad (8.19)$$

En raison de la présence du coefficient Δx^2 devant la dérivée d'ordre quatre, ce terme ne peut pas être négligé et doit être évalué au moins avec un schéma d'ordre deux. Cependant la totalité de l'approximation correspond à une précision d'ordre quatre. La dérivée quatrième peut être évaluée en utilisant l'équation de Poisson originale, $-\nabla^2\psi = \omega$, et deux différenciations successives par rapport à x , c'est-à-dire, $\partial_{xx}\partial_{xx}\psi = -\partial_{xx}\partial_{yy}\psi - \partial_{xx}\omega$. En remplaçant ∂_{xx} par δ_x^2 , on trouve :

$$\left(\delta_x^2 + \frac{\Delta x^2}{12} \delta_x^2 \partial_{yy} + \partial_{yy}\right)\psi = -\omega - \frac{\Delta x^2}{12} \delta_x^2 \omega \quad (8.20)$$

En appliquant la transformée de Fourier à l'équation (8.20) dans la direction y et en remplaçant la dérivée seconde par $-k_y'^2 \hat{\psi}$ dans l'espace de Fourier, on obtient

$$(\delta_x^2 - \frac{\Delta x^2}{12} \delta_x^2 k_y'^2 - k_y'^2) \hat{\psi} = -\hat{\omega} - \frac{\Delta x^2}{12} \delta_x^2 \hat{\omega} \quad (8.21)$$

Orlandi (2000) [77] propose de remplacer le nombre d'onde exact par le nombre d'onde modifié $k_y'^2$ qui permet d'adapter l'approximation spectrale de la dérivée seconde à la méthode aux différences finies considérée. Pour un schéma aux différences finies explicite d'ordre quatre, en se référant à la Table A.1 de l'annexe A on a $a = 4/3$, $b = -1/3$ et $\alpha = \beta = c = 0$. Donc en remplaçant les coefficients dans la relation analytique du nombre d'onde modifié donné par Lele (1992) [55], le nombre d'onde modifié pour la dérivée seconde considérée est donné par la relation suivante :

$$k_y'^2 = \frac{1}{\Delta y^2} \left[\frac{8}{3} \left(1 - \cos\left(\frac{k_y \pi}{N_y}\right) \right) - \frac{1}{6} \left(1 - \cos\left(\frac{2k_y \pi}{N_y}\right) \right) \right] \quad (8.22)$$

Le système tri-diagonal à résoudre (cf. annexe F) dans l'espace de Fourier pour chaque nombre d'onde de $\hat{\psi}$ dans la direction y est :

$$\boxed{\beta \hat{\psi}_{i+1,m} - (2\beta + k_y'^2) \hat{\psi}_{i,m} + \beta \hat{\psi}_{i-1,m} = -(\hat{\omega}_{i+1,m} + 10 \hat{\omega}_{i,m} + \hat{\omega}_{i-1,m})/12} \quad (8.23)$$

pour $i = 2, \dots, N_x - 1$, où $\beta = \Delta x^{-2} - k_y'^2/12$. Les opérations dans les directions x et y sont découplées, voir la Figure 8.2. En résumé, on applique d'abord une FFT directe 1D à terme de forçage ω dans la direction y . Ensuite, pour chaque ligne dans la direction x , le système tri-diagonal (8.23) est résolu, pour trouver la solution dans l'espace de Fourier. Finalement, pour revenir à l'espace physique, on applique une FFT inverse dans la direction y à la solution, ligne par ligne. Pour les données réelles avec conditions aux limites nulles sur les bords (Dirichlet homogène, i.e., $\psi = \omega = 0$, correspondant à des conditions aux limites d'imperméabilité et de glissement), la transformée de Fourier naturelle à utiliser est la transformée en *sinus* [56], présentée dans l'annexe D.

8.5 Modèle cinématique de nage anguilliforme

Dans cette étude, la déformation du corps du poisson est imposée. Le corps est représenté par un domaine discrétisé sur un maillage lagrangien structuré, voir la Figure 8.5. Il interagit avec le fluide environnant en exerçant des forces hydrodynamiques et des moments, calculés sur le maillage eulérien de référence. Suivant Carling et al. (1998) [67] la forme

générique du poisson repose sur un profil défini par :

$$w(s) = \begin{cases} \sqrt{2w_h\xi - \xi^2} & 0 \leq \xi < s_b \\ w_h - (w_h - w_t)\left(\frac{\xi - s_b}{s_t - s_b}\right)^2 & s_b \leq \xi < s_t \\ w_t \frac{L - \xi}{L - s_t} & s_t \leq \xi \leq L \end{cases} \quad (8.24)$$

où L est la longueur du poisson, $w_h = s_b = 0.04L$, $s_t = 0.95L$ et $w_t = 0.01L$. Le profil générique (8.24) est illustré sur la Figure 8.3. Ce profil se déforme avec une courbure imposée suivant sa ligne de symétrie, i.e., la colonne vertébrale du poisson. La longueur L du poisson reste constante. La cinématique de nage pour la majorité des poissons de

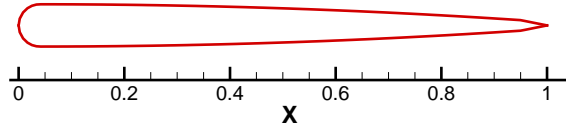


Figure 8.3: Profil du poisson donné par l'équation (8.24) avant déformation.

type *anguilliforme* et *carangiforme* peut être modélisée par la déformation de la colonne vertébrale suivant un mouvement sinusoïdal :

$$y(x, t) = a(x) \sin(2\pi(x/\lambda + ft)) \quad (8.25)$$

où λ est la longueur d'onde de la déformation imposée, f représente la fréquence de battement de la colonne vertébrale et l'enveloppe $a(x)$ est donnée dans le repère cartésien par un polynôme du second degré :

$$a(x) = a_0 + a_1x + a_2x^2 \quad (8.26)$$

Un modèle de déformation de la colonne vertébrale selon l'équation (8.25) est montré sur la Figure 8.4. Ce mode primaire du mouvement du poisson, est considéré comme le mode de propulsion. Dans la Section 8.7 on définira le deuxième mode de mouvement qui servira pour changer la direction de la nage. La longueur d'onde de déformation du poisson est définie dans le repère cartésien. Par la suite, on utilisera la théorie exacte des poutres non-linéaires pour la quantification de la cinématique du poisson. Par conséquent à la place des coordonnées dans le repère cartésien, la courbure ponctuelle de la colonne vertébrale est la seule nécessaire. La dérivée seconde de l'équation (8.25) nous donne la courbure due au mode de propulsion :

$$k_{\text{prop}}(\xi, t) = (2a_2 - (2\pi/\lambda)^2 a(\xi)) \sin(2\pi(\xi/\lambda + ft)) + (4\pi(a_1 + 2a_2\xi)/\lambda) \cos(2\pi(\xi/\lambda + ft)) \quad (8.27)$$

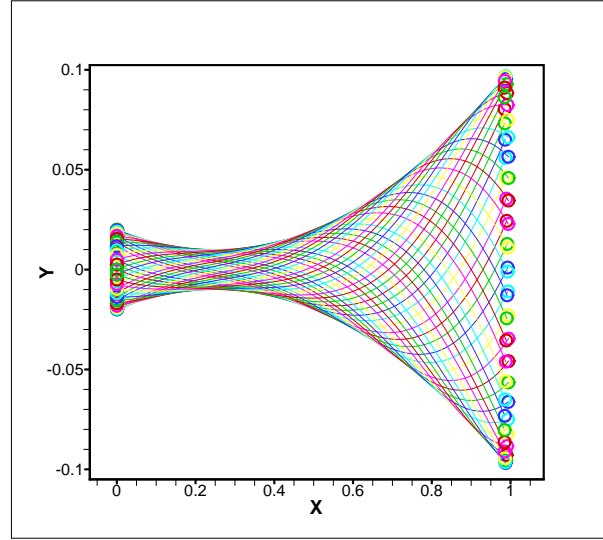


Figure 8.4: Modèle de déformation de la colonne vertébrale selon l'équation (8.25) pendant une période avec $a_0 = 0.02$, $a_1 = -0.08$, $a_2 = 0.16$, $L = 1$ et $\lambda = -1.5$.

où $a(\xi) = a_0 + a_1\xi + a_2\xi^2$. L'utilisation de la courbure du poisson constitue un cadre général pour décrire le mouvement du poisson parce que la courbure est indépendante de la direction. La technique est donc particulièrement efficace pour représenter la colonne vertébrale du poisson lorsqu'il tourne.

La théorie exacte des poutres non-linéaires a été développée par Simo (1985) [45] et a été prolongée pour la colonne vertébrale des poissons qui nagent, par Boyer et al. (2006) [106]. Dans cette théorie, la poutre est considérée comme un assemblage de sections rigides avec une épaisseur infinitésimale, i.e., un milieu unidimensionnel de Cosserat. Suivant les études de Boyer (2006) [106], Rafei et al. (2008) [120] et Belkhiri (2013) [158] en commençant par l'état de la tête comme condition de bord, la cinématique de la colonne vertébrale peut être déterminée par intégration sur la longueur de l'arc $\xi \in [0, L]$ suivant la colonne vertébrale du poisson. La variation de l'orientation en utilisant les quaternions, est donnée par :

$$\frac{\partial Q}{\partial \xi} = \frac{1}{2} M^\vee(K) Q \quad \xi \in [0, L] \quad (8.28)$$

où $Q = (\cos \frac{\phi}{2}, a_x \sin \frac{\phi}{2}, a_y \sin \frac{\phi}{2}, a_z \sin \frac{\phi}{2})^T$ sont les vecteurs unitaires, $(q_0^2 + q_1^2 + q_2^2 + q_3^2)^{1/2} = 1$ des quaternions qui représentent l'orientation du référentiel attaché au corps par rapport au référentiel inertiel (galiléen) et $M^\vee(K)$ est un tenseur anti-symétrique

$$M^\vee(K) = \begin{bmatrix} 0 & -k_1 & -k_2 & -k_3 \\ k_1 & 0 & k_3 & -k_2 \\ k_2 & -k_3 & 0 & k_1 \\ k_3 & k_2 & -k_1 & 0 \end{bmatrix} \quad (8.29)$$

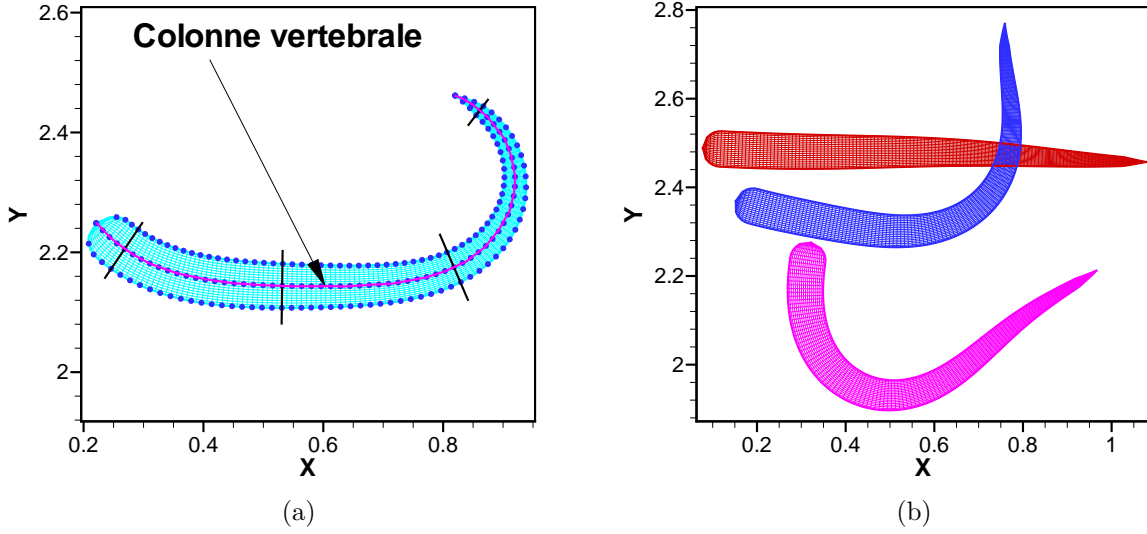


Figure 8.5: (a) Étapes de constructions du maillage structuré avec les lignes normales à la colonne vertébrale sur chaque point discret. (b) Maillages lagrangiens structurés (mobiles et déformables) qui recouvrent le poisson nageant.

où k_2 et k_3 dans $K = (k_1, k_2, k_3)^T$ représentent la courbure transversale de la colonne vertébrale et k_1 représente le taux de rotation (twist) de la section autour de la colonne vertébrale avec une normale alignée sur la direction ξ . La géométrie $R = (x, y, z)^T$ dans le référentiel galiléen, le long de la colonne vertébrale est donnée par :

$$\frac{\partial R}{\partial \xi} = Rot(Q) \Gamma \quad \xi \in [0, L] \quad (8.30)$$

où $\Gamma = (\gamma_1, \gamma_2, \gamma_3)^T$ représente le cisaillement transversal local des sections dont la première composante γ_1 est le taux d'étirement le long de la direction ξ . La matrice de rotation $Rot(Q)$ basée sur les quaternions est donnée par :

$$Rot(Q) = 2 \begin{bmatrix} q_0^2 + q_1^2 - \frac{1}{2} & q_1 q_2 - q_0 q_3 & q_1 q_3 + q_0 q_2 \\ q_1 q_2 + q_0 q_3 & q_0^2 + q_2^2 - \frac{1}{2} & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_2 q_3 + q_0 q_1 & q_0^2 + q_3^2 - \frac{1}{2} \end{bmatrix} \quad (8.31)$$

Les variations des vitesses linéaires $V = (v_1, v_2, v_3)^T$ et angulaires $\Omega = (\omega_1, \omega_2, \omega_3)^T$ moyennes dans le référentiel local, i.e., le référentiel attaché au corps, sont données par:

$$\frac{\partial}{\partial \xi} \begin{bmatrix} V \\ \Omega \end{bmatrix} = - \begin{bmatrix} K^\vee & \Gamma^\vee \\ 0 & K^\vee \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix} + \begin{bmatrix} \dot{\Gamma} \\ \dot{K} \end{bmatrix} \quad \xi \in [0, L] \quad (8.32)$$

où $\dot{\Gamma}$ et \dot{K} représentent les dérivées temporelles de Γ et K . Dans l'équation (8.32), K^\vee et Γ^\vee sont des matrices anti-symétriques construites à partir des vecteurs donnés, par exemple:

$$K^\vee = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \quad (8.33)$$

L'accélération peut être déduite par dérivation temporelle de l'équation (8.32). Pour plus de détails, on se référera à Boyer (2006) [106], Rafei et al. (2008) [120] et Belkhiri (2013). Pour trouver les vitesses dans le référentiel attaché au corps, à partir des vitesses représentées dans le référentiel galiléen et l'inverse, on utilise :

$$(v_1, v_2, v_3)^T = Rot^T(v_x, v_y, v_z)^T \quad (8.34)$$

Avec N ($1, \dots, N_{points}$) points discrets sur la colonne vertébrale du poisson, les équations (8.28), (8.30) and (8.32) doivent être intégrées simultanément le long de la colonne vertébrale par une méthode de type Runge–Kutta (en 3D on obtient $N_{eq} = 13$ équations).

Après détermination de la géométrie et des vitesses de la colonne vertébrale, un maillage lagrangien structuré se forme avec les lignes normales à la colonne vertébrale sur chaque point discret et par l'épaisseur donnée par l'équation (8.24). Des exemples sont donnés sur les Figure 8.6 et Figure 8.5. Les composantes des vitesses V_{corps} de chaque point sur le maillage lagrangien structuré avec les indices (I, J) sont données par

$$\vec{V}_{corps}(I, J) = \vec{V}_{CV}(I) + \vec{\Omega}_{CV}(I) \times \mathbf{r}(I, J) \quad (8.35)$$

où \vec{V}_{CV} et $\vec{\Omega}_{CV}$ sont les vitesses linéaire et angulaire de la colonne vertébrale, données par l'équation (8.32). Il faut s'assurer de ne pas ajouter de forces et moments artificiels. Pour cela Bergmann et Iollo (2011) [145] proposent de

1. Générer une déformation choisie.
2. Soustraire le déplacement du centre de gravité (cg).
3. Effectuer une rotation de l'opposé de l'angle induit par la déformation autour du cg.

Autrement dit en absence des forces et de moments hydrodynamiques, le déplacement du centre de gravité (cg) et la rotation du corps autour de cg (et donc les vitesses linéaires et angulaires) en raison de déformation du corps sont égales à zéro. Ensuite, les vitesses évaluées sur le maillage lagrangien doivent être transférées sur le maillage eulérien par interpolation. Dans le cadre de cette étude, on utilise une interpolation linéaire du type $\mathbf{u}_B(i, j) = \mathbf{a}xy + \mathbf{b}x + \mathbf{c}y + \mathbf{d}$. Les coefficients sont déterminés en utilisant les données des 4 plus proches points de $\mathbf{x}(i, j)$ sur le maillage lagrangien. Il suffit d'avoir l'inverse de la matrice de Vandermonde, pour calculer les coefficients de u_B et v_B .

$$\left(\begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix} \sqcup \begin{bmatrix} a_2 \\ b_2 \\ c_2 \\ d_2 \end{bmatrix} \right) = \underbrace{\begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \\ 1 & x_3 & y_3 & x_3 y_3 \\ 1 & x_4 & y_4 & x_4 y_4 \end{bmatrix}^{-1}}_{\text{matrice de Vandermonde}} \cdot \underbrace{\left(\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \sqcup \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \right)}_{\text{les vitesses du corps}} \quad (8.36)$$

où \sqcup représente le choix d'un des vecteurs entre les parenthèses et (\cdot) est le produit matriciel.

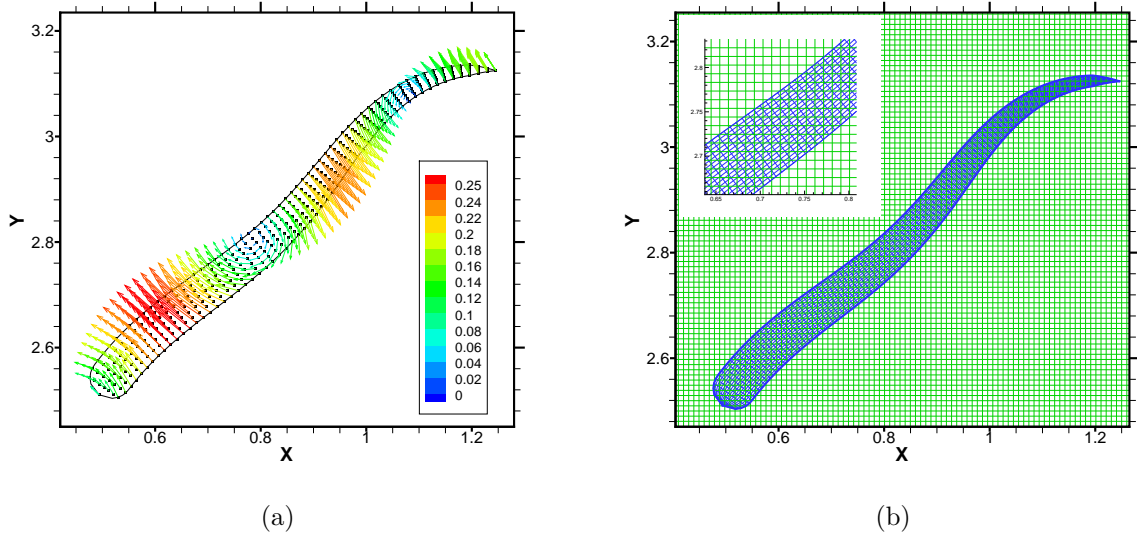


Figure 8.6: (a) Maillage lagrangien structuré couvrant le poisson en déformation et les vitesses correspondantes de chaque point, colorées suivant l'intensité de la vitesse (absolue) $\sqrt{u^2 + v^2}$. (b) Maillage lagrangien structuré composé de $Im_b \times Jm_b = 121 \times 19$ points.

8.6 Algorithme d'interaction fluide-structure

Dans cette étude, l'algorithme 4 est proposé pour le traitement de l'interaction fluide-structure pour les écoulements bidimensionnels. La validation de l'algorithme a été faite dans le Chapitre 3. L'organigramme en résumé est décrit sur la Figure 8.7.

8.7 Changement de direction du poisson

Un poisson anguilliforme peut nager vers un objectif prescrit, grâce au changement de courbure moyenne de sa colonne vertébrale. Il utilise son corps comme un gouvernail pour tourner. Dans le cadre de cette étude, l'objectif du poisson consiste à nager vers un objet, situé en un point fixe qui est prédéfini dans le domaine physique. On propose une loi efficace pour contrôler la courbure de la colonne vertébrale d'un poisson lorsque celui-ci souhaite

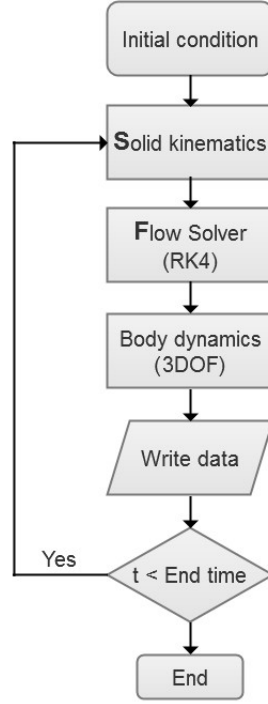


Figure 8.7: Organigramme de l'algorithme d'interaction fluide-structure.

effectuer un changement de direction. Pour atteindre un objectif prédéfini, à notre modèle il convient d'ajouter une courbure constante $k_{\text{offset}}(t)$ le long de la colonne vertébrale du poisson $\xi \in [0, l_{\text{fish}}]$, sur son mode de propulsion primaire, i.e.,

$$k_3 = k_{\text{prop}}(\xi, t) + k_{\text{offset}}(t) \quad (8.37)$$

où $k_{\text{prop}}(\xi, t)$ est la courbure due au mouvement sinusoïdal donné par l'équation (8.27). Afin d'effectuer un mouvement physiquement raisonnable, le changement Δk de courbure rajoutée k_{offset} suivant l'équation (8.39) doit être intégré progressivement au cours de temps, i.e., à l'ordre $O(\Delta t)$. Pour un poisson qui nage tout droit, k_{offset} est égal à zéro. Pour effectuer un changement de direction, une courbure désirée k_{des} doit être estimée avec la relation suivante,

$$k_{\text{des}}(\theta_{\text{des}}) = \begin{cases} -\text{sgn}(\theta_{\text{des}}) k_{\text{max}} & |\theta_{\text{des}}| \geq \theta_{\text{limit}} \\ -\text{sgn}(\theta_{\text{des}}) k_{\text{max}} \left(\frac{\theta_{\text{des}}}{\theta_{\text{limit}}}\right)^2 & \text{sinon} \end{cases} \quad (8.38)$$

où sgn représente la fonction signe, i.e., $\text{sgn}(\theta_{\text{des}}) = \theta_{\text{des}}/|\theta_{\text{des}}|$, ici θ_{limit} est la limite de la région de croissance hyperbolique de $k_{\text{des}}(\theta_{\text{des}})$, voir la Figure 8.8 (b). Les valeurs positives et négatives de θ_{des} (dans le référentiel attaché au corps) conduisent le poisson à tourner à gauche et à droite, respectivement. Pour une représentation schématique de l'angle désiré θ_{des} , il convient de se reporter à la Figure 8.9.

A chaque pas de temps, l'angle désiré θ_{des} doit être d'abord calculé selon la position et la

direction de la tête tout en visant l'objectif. Après cela, en utilisant l'équation (8.38), une courbure désirée k_{des} doit être estimée. Ensuite, k_{offset} peut être calculé avec la relation suivante,

$$k_{\text{offset}}^{n+1}(k_{\text{des}}) = \begin{cases} k_{\text{offset}}^n + \Delta k & k_{\text{offset}}^n < k_{\text{des}} \\ k_{\text{offset}}^n - \Delta k & k_{\text{offset}}^n > k_{\text{des}} \\ k_{\text{offset}}^n & k_{\text{offset}}^n = k_{\text{des}} \end{cases} \quad (8.39)$$

où $\Delta k = \Delta t \pi / T$. Finalement suivant l'équation (8.37), on rajoute k_{offset} à la courbure initiale k_{prop} de la colonne vertébrale, afin que le poisson réalise un changement de direction pendant son déplacement. En résumé connaissant la direction, la position et les vitesses linéaire et angulaire de la tête du poisson les équations (8.28), (8.30) et (8.32) doivent être intégrées simultanément le long du poisson pour trouver les positions et les vitesses de la colonne vertébrale.

Dans le cas de la nage anguilliforme la longueur du poisson est constante, donc sur la colonne vertébrale on a un taux d'étirement constant et le cisaillement local égal à zéro, i.e., $\Gamma = (1, 0, 0)$. Dans le cas de la nage bidimensionnelle on a qu'une seule courbure à imposer, i.e., k_3 , le taux de rotation (twist) et la courbure transversale sont égales à zéro, par conséquent on a $K = (0, 0, k_3)$. On considère $Im_b = 251$ points discrets sur la colonne vertébrale du poisson et $Jm_b = 39$ points dans la direction latérale, pour construire le maillage lagrangien qui couvre le poisson. En choisissant $k_{\text{max}} = \pi$ dans l'équation (8.38) le poisson prend la forme d'un demi-cercle quand il tourne avec sa courbure maximale. Comme dans les études de Bergmann et Iollo (2011) [145] on utilise $\theta_{\text{limit}} = \pi/4$, voir la Figure 8.8 (b). La dérivée temporelle de la courbure dk/dt est nécessaire dans l'équation (8.32) pour calculer les vitesses et elle peut être estimée numériquement.

Une simulation est effectuée pour montrer la performance de l'algorithme proposé, qui mène le poisson vers son objectif de position prédéfinie. La taille du domaine de la solution est $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, le maillage eulérien est composé de 1024×1024 points, le paramètre de pénalisation est $\eta = 10^{-3}$, la bande de filtrage des coefficients hydrodynamiques est égale à $\delta = 0.005$, la fréquence de battement du poisson est choisi égal à $f = 1$ et la longueur d'onde de déformation de son corps est $\lambda = 1$. L'enveloppe de la colonne vertébrale est paramétrée avec $a_2 = 0$, $a_1 = 0.125/(1 + c)$, $a_0 = 0.125c/(1 + c)$ et $c = 0.03125$.

Le profil du poisson est donné par l'équation (8.24) et il est illustré sur la Figure 8.3. La viscosité cinématique utilisée est $\nu = 1.4 \times 10^{-4}$, la position initiale de la tête située à $(x_0, y_0) = (0.1L_x, 0.5L_y)$ et l'angle initial de la tête est égal à $\theta_0 = 0$. La Figure 8.10 (a-f) montre quelques distributions instantanées des champs de vorticit  , obtenus lors de la simulation de la nage du poisson consid  r  e vers un objectif pr  d  fini situ      $(x_f, y_f) = (0.9L_x, 0.5L_y)$. La simulation commence avec le corps et le fluide environnant au repos, i.e., $\mathbf{u}_B(\mathbf{x}, 0) = 0$ et $\omega(\mathbf{x}, 0) = \psi(\mathbf{x}, 0) = 0$. Des conditions aux limites d'imperm  abilit   et de glissement (Dirichlet homog  ne, i.e., $\psi|_{\partial\Omega} = \omega|_{\partial\Omega} = 0$) sont im-

posées sur les bords.

Le mise en mouvement du poisson s'accompagne d'un accroissement progressif de la courbure de la colonne vertébrale selon l'équation (8.37) au cours de la première période T . Du fait de la fonction sinusoïdale modèle qui est tracée sur la Figure 8.8 (a), la courbure s'accroît progressivement de zéro jusqu'à sa valeur prévue. A proximité de l'objectif ($r_{\text{objectif}} = 0.5l_{\text{poisson}}$), la courbure de la colonne vertébrale du poisson, donnée par l'équation (8.37), tend vers zéro (voir les Figures 8.10 et 8.12) en la multipliant par la fonction suivante,

$$C(t) = \frac{t_f - t}{t_f - t_i} + \frac{1}{2\pi} \sin(2\pi \frac{t - t_i}{t_f - t_i}) , \quad t \in [t_i, t_f] \quad (8.40)$$

qui est l'inverse de la fonction présentée sur la Figure 8.8 (a), avec $t_i = t_{\text{arriver}}$, $t_f = t_{\text{arriver}} + T$ pour faire décroître progressivement la courbure de la colonne vertébrale au cours d'une période. Les positions successives du poisson représentées par sa colonne vertébrale au cours de sa nage vers son objectif prédéfini situé à $(x_f, y_f) = (0.9L_x, 0.5L_y)$, sont représentées pour un intervalle de temps $t \in [0, 15]$ sur la Figure 8.12. On peut constater sur la Figure 8.10 (a-f) que les valeurs du champ de vorticité initialement égales à zéro (correspondant au repos du poisson et du fluide environnant) s'accroissent rapidement $\omega \in [-200, 220]$ au cours du changement de direction. Lors de la nage en ligne droite, les valeurs du champ de vorticité oscillent dans la gamme de $\omega \in [-60, 70]$. Finalement à proximité de l'objectif, l'amplitude du battement du corps se réduit suivant l'équation (8.40), les valeurs du champ de vorticité décroissent pour être dans la gamme de $\omega \in [-28, 25]$.

Un point selle dans l'écoulement correspond à une stagnation de fluide par suite de collision de deux courants. La succession des points selles et des centres (correspondant aux centres des tourbillons) qui se manifestent dans l'écoulement autour du poisson, est illustrée sur la Figure 8.11. Ce sont les caractéristiques communes des écoulements séparés sur un corps. Pour calculer le champ de pression, une équation de Poisson est extraite des équations de quantité de mouvement (8.1) :

$$\nabla \cdot (\nabla p) = -\rho_f \nabla \cdot [(\mathbf{u} \cdot \nabla) \mathbf{u}] - \rho_f \nabla \cdot \mathbf{F} \quad (8.41)$$

Pour l'algorithme proposé, elle peut se simplifier selon la forme suivante :

$$\nabla^2 p = 2\rho_f(u_x v_y - u_y v_x) - \rho_f \nabla \cdot [\eta^{-1} \chi(\mathbf{u} - \mathbf{u}_p)] \quad (8.42)$$

où les conditions aux limites de type Neumann $\partial p / \partial \mathbf{n}|_{\partial \Omega} = 0$ sont imposées sur les bords du domaine rectangulaire. En utilisant une méthode aux différences finies d'ordre deux, sur la paroi gauche on a :

$$p_1 = (4p_2 - p_3)/3$$

Des relations similaires aux frontières vers l'arrière-aval peuvent être déduites pour les quatre bords du domaine. L'équation de Poisson (8.42) après discrétisation conduit à un système d'équations linéaires. On utilise la méthode de sur-relaxation successive, qui est une variante de la méthode de Gauss-Seidel, pour résoudre le système d'équations linéaires [56]. Grâce à un balayeur de type échiquier (rouge et noir) l'efficacité de la méthode de Gauss-Seidel peut encore être accrue, parce que les valeurs les plus récentes sont toujours utilisées au cours des itérations. Dans cette étude le champ de pression est calculé une fois toutes les 500 itérations. En effet, la pression n'est plus présente dans l'algorithme et ne servira que pour les analyses nécessaires et la visualisation. En présence des conditions aux bords de type Neumann, les solutions de l'équation de Poisson sont singulières (multiples), i.e., définies à une constante près. Pour éviter cela, dans les solutions de l'équation de Poisson, au cours des itérations la valeur de la pression au centre du domaine est forcée à une valeur constante $p_{\text{centre}} = \text{cte}$, i.e.,

$$p(N_x/2, N_y/2) = 1$$

Quelques représentations instantanées des champs de pression sont montrées sur la Figure 8.10 (g-l). Les régions de surpression et de dépression sont alternativement visibles de part et d'autre du poisson lors de son avancement. Comme prévu, les isovaleurs de pression sont normales à la surface du poisson. Dans l'écoulement, les centres des tourbillons correspondent à des régions de dépression. La déviation du champ de pression par rapport à $p_\infty = 1$ s'accroît à $p \in [-21, 27]$ une fois que le battement commence et décroît très vite, lorsque à $t = 15$ le battement s'arrête à proximité de l'objectif. Ceci est en opposition claire avec le champ de vorticité qui perdure même après l'arrêt du poisson, ce qui démontre la nature elliptique du champ de pression vis-à-vis de la nature parabolique du champ de vorticité. Une région de surpression est visible entre la tête et la queue du poisson quand celui-ci tourne en configurant un demi-cercle avec la courbure maximum rajoutée, i.e., $k = \pi$, qui correspond également à ce qu'a constaté Gazzola et al. (2012) [155].

Comme démontré par Gazzola et al. (2012) [155], la manœuvre de retournement en fer à cheval est un mécanisme efficace pour changer la direction de la nage et repartir en même temps. En prenant la forme d'un demi-cercle, les larves stockent un volume considérable de fluide qu'elles éjecteront pour accélérer et repartir. Quand des objets rigides font face à un écoulement, comme dans le cas du transport industriel (train, avion, voiture, ...), la pression maximum se manifeste au point de stagnation du front. Dans le cas de la nage des poissons, contrairement au déplacement classique de corps rigides, le point de stagnation ne se trouve plus au nez de l'objet. Lors de la nage, les régions de surpression et de dépression se développent de part et d'autre du poisson, engendrant des points de stagnation tout le long du profil du poisson. Dans l'étape finale de la nage après l'arrêt du battement à $t = 15$, comme le montre la Figure 8.10 (l), une zone de surpression réapparaît devant la

tête du poisson.

Le lissage de la fonction caractéristique $\bar{\chi}$ par l'équation (8.7), génère des champs de pression réguliers. Aucune oscillation du champ de pression à l'intérieur et à l'extérieur du poisson n'est visible. Grâce au modèle de rotation proposé qui superpose une courbure en fonction de temps (constant le long du poisson) sur le mode sinusoïdal de propulsion, le poisson à l'instant du retournement effectue un changement de direction de 180° dans une aire qui correspond à environ 1.3 fois sa longueur.

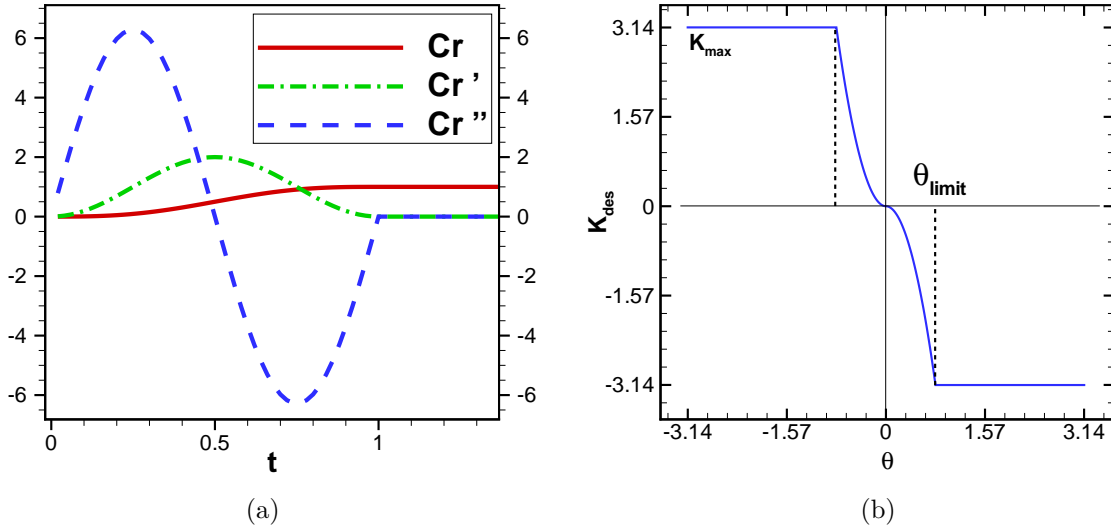


Figure 8.8: (a) Fonction proposée par Boyer et al. (2006) [106] pour accroître progressivement la courbure de la colonne vertébrale du poisson : $Cr(t) = t' - \sin(2\pi t')/(2\pi)$, $t \in [t_i, t_f]$ avec $t' = (t - t_i)/(t_f - t_i)$, $t_i = 0$ et $t_f = 1$. A $t = 0$ et $t = 1$ les limites à gauche et les limites à droite sont égales pour la fonction Cr et pour ses dérivées première Cr' et seconde Cr'' . (b) Fonction proposée pour estimer la courbure désirée $k_{des}(\theta)$ suivant l'équation (8.38) avec $k_{max} = \pi$ et $\theta_{limit} = \pi/4$.

8.8 Conclusion

Dans cette étude, une méthode numérique précise et efficace est proposée pour la simulation de corps déformables interagissant avec un écoulement incompressible. Les équations de Navier–Stokes, considérées dans leur formulation vorticité-fonction de courant, sont discrétisées temporellement et spatialement à l'aide respectivement d'un schéma d'ordre quatre de Runge–Kutta et par des différences finies compactes. Grâce à l'utilisation d'un maillage uniforme, nous proposons un nouveau “solveur direct” au quatrième ordre pour l'équation de Poisson, permettant de garantir la contrainte d'incompressibilité au niveau du zéro machine sur une grille optimale. L'introduction d'un corps déformable et mobile dans l'écoulement de fluide est réalisée au moyen d'une méthode de pénalisation de volume.

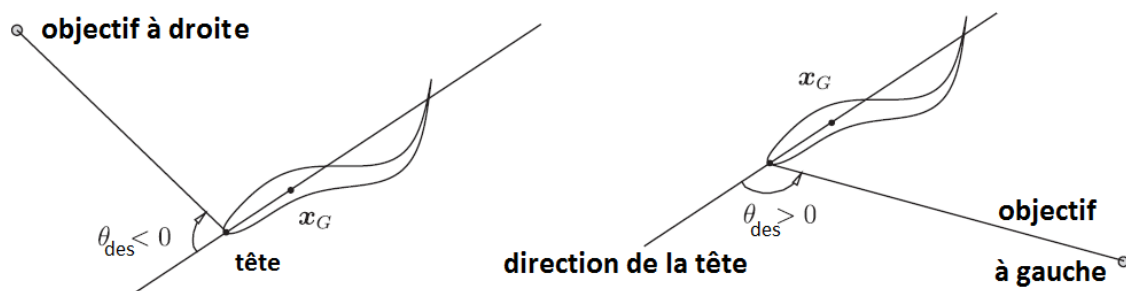


Figure 8.9: Une représentation schématique de l'angle désiré pour contrôler la courbure rajoutée k_{offset} (le long de la colonne vertébrale) du poisson afin de le diriger vers son objectif. Ici $\theta_{des} = \theta_{objectif} - \theta_{tete}$ est l'angle entre la direction de la tête et la ligne reliant la tête à la position de son objectif, $(-\pi < \theta_{des} < \pi)$. Image adoptée d'après Bergmann et Iollo (2011) [145] avec quelques modifications.

La déformation du corps est imposée par l'utilisation d'un maillage lagrangien structuré et mobile qui interagit avec le fluide environnant en produisant des forces hydrodynamiques et des moments (calculés sur le maillage eulérien de référence). Une loi efficace de contrôle de la courbure du poisson anguilliforme nageant vers son objectif prescrit, est proposée. Grâce à ce modèle, le poisson initialement au repos commence à nager avec un changement de direction complet dans une aire réduite correspondant à seulement 1.3 fois sa longueur. La loi de contrôle de la courbure est basée sur la théorie exacte des poutres non-linéaires. Ensuite pour augmenter l'efficacité de la méthode, dans le Chapitre 5 une analyse multiéchelle est appliquée à l'algorithme, permettant de réduire significativement le nombre de points nécessaires. La grille se raffine automatiquement dans les régions présentant un fort gradient. La stratégie d'adaptation est basée sur la transformée en ondelettes puis le seuillage des coefficients. Les résultats obtenus montrent que le temps de calcul peut être réduit considérablement avec la méthode multiéchelle tout en conservant la précision. Finalement, dans le Chapitre 6, une simulation de nage tri-dimensionnelle a été réalisée avec la méthode de pénalisation de volume appliquée au code Incompact3d, développé par Laizet et Lambalais (2009) [129], qui est en accès libre. La méthode numérique développée prouve son efficacité et sa précision tant dans le cas de la nage du poisson que dans le cas d'autres problèmes d'interactions fluide-structure comme la sédimentation d'un cylindre ou d'une ellipse. Le code est librement accessible et a été développé en FORTRAN [171].

Algorithm 4 L'algorithme d'interaction fluide-structure

1. INTRODUCTION D'UN ÉTAT INITIAL

2. CINÉMATIQUE DU CORPS

- (a) (Spécifiquement pour le poisson) Construction de la colonne vertébrale du poisson par l'intégration des équations (8.28), (8.30) et (8.32)
- (b) (Spécifiquement pour le poisson) Recouvrement du poisson par un maillage lagrangien structuré, puis calcul des vitesses en chaque point du corps déformable avec l'équation (8.35). S'assurer de ne pas ajouter de forces et moments artificiels.
- (c) Calcul de la fonction caractéristique $\chi(i, j)$ et lissage avec l'équation (8.7)
- (d) Calcul du moment d'inertie I_{zz} autour du point de référence
- (e) (Spécifiquement pour le poisson) Transformation des vitesses du corps sur le maillage eulérien par interpolation (8.36) pour obtenir $u_B(i, j)$ et $v_B(i, j)$. (Lagrange \rightarrow Euler)

3. INTÉGRATION TEMPORELLE DE L'ÉCOULEMENT AU MOYEN DU SCHÉMA RK4

- (a) $\omega_0 = \omega^n$, $\psi_0 = \psi^n$
Pour $i = 1, 2, 3$ ($\alpha_1 = \alpha_2 = 1/2$ et $\alpha_3 = 1$)
- (b) Calcul $k_i(\omega, \psi)^{i-1} = -\partial_y \psi \partial_x \omega + \partial_x \psi \partial_y \omega + \nu \nabla^2 \omega + \partial_x F_y - \partial_y F_x$
- (c) $\omega_i = \omega^n + \alpha_i \Delta t k_i$
- (d) Résolution de l'équation (8.4); $-\nabla^2 \psi_i = \omega_i$ pour mise à jour des vitesses (u, v)
Fin pour $i = 1, 2, 3$
- (e) Calcul $k_4(\omega, \psi)^3 = -\partial_y \psi \partial_x \omega + \partial_x \psi \partial_y \omega + \nu \nabla^2 \omega + \partial_x F_y - \partial_y F_x$
- (f) Mise à jour de la vorticité ; $\omega^{n+1} = \omega^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
- (g) Résolution de l'équation (8.4); $-\nabla^2 \psi^{n+1} = \omega^{n+1}$

4. DYNAMIQUE DU CORPS

- (a) Calcul des coefficients hydrodynamiques du corps avec les équations (8.8) et (8.9)
- (b) Débruitage des coefficients hydrodynamiques avec l'équation (8.10)
- (c) Calcul des déplacements avec l'équation (8.13) et les vitesses linéaires
- (d) Calcul du mouvement de rotation avec l'équation (8.15) et la vitesse angulaire

5. ÉCRITURE DES DONNÉES NÉCESSAIRES DANS LES FICHIERS

6. SI $T < T_{end}$, REPRENDRE L'ÉTAPE 2

7. FIN

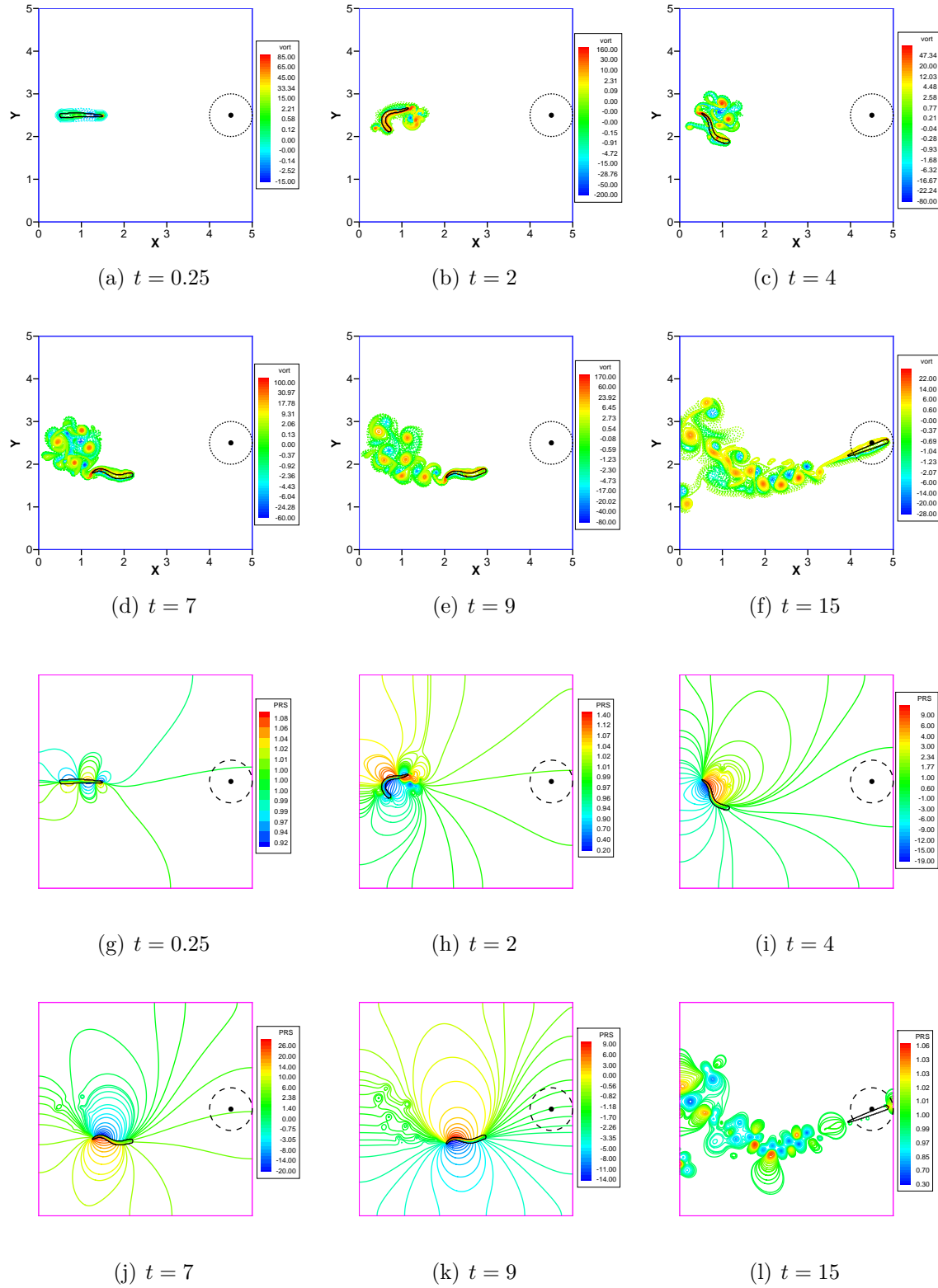


Figure 8.10: Champs de vorticit  (a-f) et de pression (g-l) autour du poisson (repr sent  par les lignes noires correspondant   $\chi = 0.2$) nageant vers un objectif pr d fini, situ  au point $(x_f, y_f) = (0.9L_x, 0.5L_y)$. A $t = 0$, le poisson et le fluide environnant sont au repos. La domaine de la solution est $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, la r solution du maillage eul rien est 1024×1024 , la r solution du maillage lagrangien est $(Im_b \times Jm_b = 251 \times 39)$, le param tre de p nalisation est $\eta = 5 \times 10^{-4}$ et la viscosit  cin matique est  gale   $\nu = 1.4 \times 10^{-4}$.

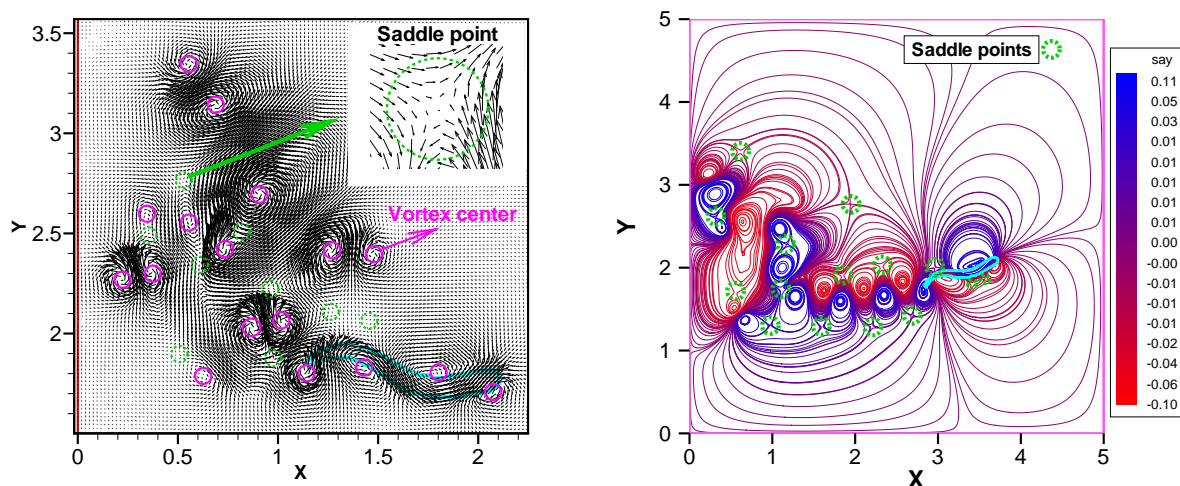


Figure 8.11: Les points selles (entourés des cercles en pointillés verts) et centres (entourés des cercles solides violets) dans l'écoulement séparé autour du poisson sont successivement lâchés par le mouvement du corps. Deux tourbillons forment un dipôle qui génère un jet localisé vers l'arrière dans l'écoulement au cours de la nage du poisson.

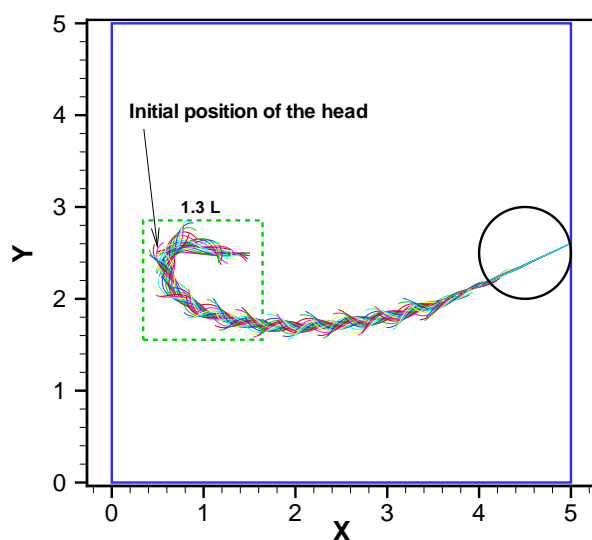


Figure 8.12: Les positions successives du poisson matérialisées par sa colonne vertébrale au cours de sa nage vers l'objectif prédéfini situé à $(x_f, y_f) = (0.9L_x, 0.5L_y)$ sont représentées pour un intervalle de temps $t \in [0, 15]$. A proximité de l'objectif ($r_{\text{objectif}} = 0.5l_{\text{poisson}}$) la courbure de la colonne vertébrale du poisson, donnée par l'équation (8.37), se ramène à zéro. Les champs de vorticit  et de pression correspondants sont illustr s sur la Figure 8.10. Le poisson initialement au repos effectue un changement de direction de 180° pr s du bord gauche du domaine dans une aire qui correspond   environ 1.3 fois sa longueur.

Appendix A

Compact differentiation

Classical finite differences are based on Lagrange interpolation. Therefore high-order approximations lead to large stencils. In compact finite differences Hermit interpolation is used to keep high accuracy and compact stencil. For a given discrete function f , on $(i = 1, 2, \dots, N)$ uniformly distributed collocated grid points with spacing $h = L/(N - 1)$, explicit or implicit schemes with different orders of accuracy for approximation of the first derivative f' are constructed by Lele [55] from the following general relation

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = a \frac{f_{i+1} - f_{i-1}}{2h} + b \frac{f_{i+2} - f_{i-2}}{4h} + c \frac{f_{i+3} - f_{i-3}}{6h} \quad (\text{A.1})$$

An approximation of the first derivative with sixth-order accuracy is obtained by

$$\frac{1}{3}f'_{i-1} + f'_i + \frac{1}{3}f'_{i+1} = \frac{14}{9} \frac{f_{i+1} - f_{i-1}}{2h} + \frac{1}{9} \frac{f_{i+2} - f_{i-2}}{4h}, \quad i = 3, \dots, N - 2 \quad (\text{A.2})$$

If fourth-order accuracy is desired one can use the classical Padé scheme:

$$\frac{1}{4}f'_{i-1} + f'_i + \frac{1}{4}f'_{i+1} = \frac{3}{2} \frac{f_{i+1} - f_{i-1}}{2h}, \quad i = 2, \dots, N - 1 \quad (\text{A.3})$$

near the boundaries a third-order accuracy can be achieved by

$$f'_1 + 2f'_2 = (-2.5f_1 + 2f_2 + 0.5f_3)/h \quad (\text{A.4})$$

$$f'_N + 2f'_{N-1} = -(-2.5f_N + 2f_{N-1} + 0.5f_{N-2})/h \quad (\text{A.5})$$

A similar relation holds for evaluation of the second derivative f''

$$\beta f''_{i-2} + \alpha f''_{i-1} + f''_i + \alpha f''_{i+1} + \beta f''_{i+2} = a \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + b \frac{f_{i+2} - 2f_i + f_{i-2}}{4h^2} + c \frac{f_{i+3} - 2f_i + f_{i-3}}{9h^2} \quad (\text{A.6})$$

An approximation of the second derivative with sixth-order accuracy is given by

$$\frac{2}{11}f''_{i-1} + f''_i + \frac{2}{11}f''_{i+1} = \frac{12}{11} \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + \frac{3}{11} \frac{f_{i+2} - 2f_i + f_{i-2}}{4h^2}, \quad i = 3, \dots, N - 2 \quad (\text{A.7})$$

If fourth-order accuracy (classical Padé scheme) is desired the following equation can be used

$$\frac{1}{10}f''_{i-1} + f''_i + \frac{1}{10}f''_{i+1} = \frac{12}{10} \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2}, \quad i = 2, \dots, N - 1 \quad (\text{A.8})$$

near the boundaries a third-order accuracy can be achieved by

$$f_1'' + 11f_2'' = (13f_1 - 27f_2 + 15f_3 - f_4)/h^2 \quad (\text{A.9})$$

$$f_N'' + 11f_{N-1}'' = (13f_N - 27f_{N-1} + 15f_{N-2} - f_{N-3})/h^2 \quad (\text{A.10})$$

The coefficients of some commonly used methods for approximation of the first f' and the second derivative f'' , with different orders of accuracy, are given in Tables A.1 - A.4.

An optimized spectral-like method introduced by Kim [111] for evaluation of the first derivative. This compact formulation leads to formally fourth-order accuracy everywhere and especially near the boundaries. The method of Kim [111] can be represented in the following matrix form:

$$[\mathbf{P}]f' = [\mathbf{Q}]f \quad (\text{A.11})$$

where $[\mathbf{P}]$ and $[\mathbf{Q}]$ represent $N \times N$ matrices. On the left-hand side, \mathbf{P} is a pentadiagonal band matrix of the form:

$$\mathbf{P} = \begin{bmatrix} 1 & \gamma_{01} & \gamma_{02} & 0 & \cdots & 0 & 0 & 0 & 0 \\ \gamma_{10} & 1 & \gamma_{12} & \gamma_{13} & 0 & \cdots & 0 & 0 & 0 \\ \gamma_{20} & \gamma_{21} & 1 & \gamma_{23} & \gamma_{24} & 0 & \cdots & 0 & 0 \\ 0 & \beta & \alpha & 1 & \alpha & \beta & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \beta & \alpha & 1 & \alpha & \beta & 0 \\ 0 & 0 & \cdots & 0 & \gamma_{24} & \gamma_{23} & 1 & \gamma_{21} & \gamma_{20} \\ 0 & 0 & 0 & \cdots & 0 & \gamma_{13} & \gamma_{12} & 1 & \gamma_{10} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \gamma_{02} & \gamma_{01} & 1 \end{bmatrix} \quad (\text{A.12})$$

On the right-hand side of (A.11), \mathbf{Q} is not strictly band limited matrix and have a form:

$$\mathbf{Q} = \frac{1}{h} \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} & b_{04} & b_{05} & b_{06} & 0 & 0 & \cdots & 0 \\ b_{10} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} & 0 & 0 & \cdots & 0 \\ b_{20} & b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} & 0 & 0 & \cdots & 0 \\ -c/6 & -b/4 & -a/2 & 0 & a/2 & b/4 & c/6 & 0 & 0 & \cdots & 0 \\ 0 & -c/6 & -b/4 & -a/2 & 0 & a/2 & b/4 & c/6 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -c/6 & -b/4 & -a/2 & 0 & a/2 & b/4 & c/6 & 0 \\ 0 & \cdots & 0 & 0 & -c/6 & -b/4 & -a/2 & 0 & a/2 & b/4 & c/6 \\ 0 & \cdots & 0 & 0 & -b_{26} & -b_{25} & -b_{24} & -b_{23} & -b_{22} & -b_{21} & -b_{20} \\ 0 & \cdots & 0 & 0 & -b_{16} & -b_{15} & -b_{14} & -b_{13} & -b_{12} & -b_{11} & -b_{10} \\ 0 & \cdots & 0 & 0 & -b_{06} & -b_{05} & -b_{04} & -b_{03} & -b_{02} & -b_{01} & -b_{00} \end{bmatrix} \quad (\text{A.13})$$

For the inner points the coefficients are given in Table A.4. For near boundary points the coefficients are listed in Table A.5. This method was originally proposed in the context of aeroacoustic by Kim [111], for evaluation of the first derivative while keeping the fourth-order accuracy near the boundaries.

Table A.1: Coefficients of explicit differentiation ($\alpha = \beta = 0$) for the first f' and the second derivative f'' with Eqs. (A.1) and (A.6).

Coefficients	2 th -order	4 th -order	6 th -order
a	1	4/3	3/2
b	0	-1/3	-3/5
c	0	0	1/10

Table A.2: Coefficients of implicit compact differentiation (via tri-diagonal system of equations, $\beta = 0$) for the first derivative f' with Eq. (A.1).

Coefficients	4 th -order (Padé)	6 th -order (Lele)	8 th -order (Lele)
α	1/4	1/3	3/8
a	3/2	14/9	25/16
b	0	1/9	1/5
c	0	0	-1/80

Table A.3: Coefficients of implicit compact differentiation (via tri-diagonal system of equations, $\beta = 0$) for the second derivative f'' with Eq. (A.6).

Coefficients	4 th -order (Padé)	6 th -order (Lele)	8 th -order (Lele)
α	1/10	2/11	9/38
a	12/10	12/11	(696 - 1191 α)/428
b	0	3/11	(2454 α - 294)/535
c	0	0	(1179 α - 344)/2140

Table A.4: Coefficients of spectral-like (formally fourth-order) implicit compact differentiation (via five-diagonal system of equations) for the first derivative f' with Eq. (A.1) for inner points.

Coefficients	4 th -order Lele [55]	4 th -order Kim [111]
α	0.5771439	0.5862704032801503
β	0.0896406	0.09549533555017055
a	1.3025166	1.2862813473838312
b	0.99355	1.0344044093980264
c	0.03750245	0.04284572087878425

Table A.5: Coefficients of implicit compact differentiation (which leads to a pentadiagonal system of linear equations) for the near boundary points for evaluation of the first derivative f' via Eq. (A.11). Optimised by Kim [111] to keep the fourth-order accuracy near the boundaries.

Coefficients	1 st -point($i = 0$)	2 nd -point($i = 1$)	3 rd -point($i = 2$)
Left			
γ_{i0}	1	0.08360703307833438	0.03250008295108466
γ_{i1}	5.912678614078549	1	0.3998040493524358
γ_{i2}	3.775623951744012	2.058102869495757	1
γ_{i3}	-	0.9704052014790193	0.771926127761586
γ_{i4}	-	-	0.16266359312569
Right			
b_{i0}	$-\sum_{j=1}^6 b_{ij}$	-0.3177447290722621	-0.1219006056449124
b_{i1}	-3.456878182643609	$-\sum_{(j=0, j \neq 1)}^6 b_{ij}$	-0.6301651351188667
b_{i2}	5.83904335883473	-0.02807631929593225	$-\sum_{(j=0, j \neq 2)}^6 b_{ij}$
b_{i3}	1.015886726041007	1.593461635747659	0.6521195063966084
b_{i4}	-0.2246526470654333	0.2533027046976367	0.393884355121035
b_{i5}	0.08564940889936562	-0.03619652460174756	0.01904944407973912
b_{i6}	-0.01836710059356763	0.004080281419108407	-0.001027260523947668

Appendix B

The volume penalization method

By using the volume penalization method for fluid interaction with deformable bodies, some variables like gravity center, moments of inertia, hydrodynamic coefficients, etc. must be evaluated at each time step. After definin the mask function $\chi(i, j, k)$ on the Eulerian grid, integral variables can be approximated numerically. Some examples are given in the following: The volume of the penalized area over a Cartesian uniform grid is determined by the following relation

$$V_{pen} = \int_V \chi \, dv \approx \Delta x \Delta y \Delta z \sum_{k=1}^{Kmax} \sum_{j=1}^{Jmax} \sum_{i=1}^{Imax} \chi_{ijk}$$

in two-dimensions the volume integrals must be replaced by surface integrals

$$S_{pen} = \int_{\Omega_s} \chi \, ds \approx \Delta x \Delta y \sum_{i=1}^{Imax} \sum_{j=1}^{Jmax} \chi_{i,j}$$

The geometrical moment of inertia, with dimension $[ML^2]$, about an axis through the center of mass is given by

$$I = \rho_s \int_V \chi \mathbf{r}^2 \, dv$$

The polar moment of inertia $I_{zz} = I_x + I_y$ around z axis (also called J_z) is defined as follows

$$I_{zz} = \rho_s \int_{\Omega_s} \chi \mathbf{r}^2 \, ds$$

where ρ_s is the density of the immersed body, I_{zz} is given with respect to the origin of the vector \mathbf{r} which is arbitrary and the z axis is passing through it. For $\rho_s = cte$ the polar moment of inertia with respect to a reference point reads

$$I_{zz}^{ref} = \rho_s \int_V \chi ((x - x_{ref})^2 + (y - y_{ref})^2) \, dv$$

in two-dimensions the volume integral must be replaced by a surface integral over Ω_s In three dimensions the moment of inertia matrix is defined as

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$$

where as an example I_{xy} is given by:

$$I_{xy}^{ref} = I_{yx}^{ref} = \rho_s \int_V \chi ((x - x_{ref})(y - y_{ref})) dv$$

other elements can be deduced in a similar way. The center of gravity is the point in a body around which the resultant torque due to gravity forces vanishes. Near the surface of the earth, where the gravity acts downwards as a parallel force field, the center of gravity and the center of mass are the same. If the mass distribution is continuous with the density $\rho_s(\mathbf{r})$ within a volume V , then the integral of the weighted position coordinates of the points in this volume relative to the center of mass \mathbf{x}_{cg} is zero, that is

$$\int_V \chi \rho_s(\mathbf{r})(\mathbf{x} - \mathbf{x}_{cg}) dv = 0$$

Solveing this equation, the coordinates of gravity center \mathbf{x}_{cg} is given by

$$\mathbf{x}_{cg} = \frac{1}{M} \int_V \chi \rho_s(\mathbf{r}) \mathbf{x} dv$$

where M is the total mass in the volume. In two-dimensions for $\rho_s = cte$, we get

$$\mathbf{x}_{cg} = \frac{1}{S} \int_S \chi \mathbf{x} ds$$

If a continuous mass distribution has uniform density, which means that ρ_s is constant, then the center of mass is the same as the centroid of the volume. On a uniform grid in two-dimensions \mathbf{F} can be numerically evaluated as

$$\mathbf{F} \approx \underbrace{\frac{\rho_f}{\eta} \Delta x \Delta y \sum_{i=1}^{Imax} \sum_{j=1}^{Jmax} \chi_{i,j} (\mathbf{u} - \mathbf{u}_B)_{i,j}}_{S_{pen}} + \rho_f S_{pen} \ddot{\mathbf{x}}_{cg} \quad (\text{B.1})$$

For a second-order discretization of the terms like $\partial_x(\chi\psi_x)$ the following conservative relation can be used:

$$\left. \frac{\partial}{\partial x} \left(\chi \frac{\partial \psi}{\partial x} \right) \right|_i = \frac{(\chi_{i+1} + \chi_i)(\psi_{i+1} - \psi_i) - (\chi_{i-1} + \chi_i)(\psi_i - \psi_{i-1})}{2h^2} + O(h)^2 \quad (\text{B.2})$$

Appendix C

The coefficient matrix of Poisson equation

An iterative method presented in Section 2.3.1 for high-order solution of the Poisson equation $\nabla^2\psi = -\omega$. A linear system of equations (2.77) is resulted from fourth-order discretization of the Poisson equation. The 9-point stencil used for discretization, is illustrated in Fig. 2.12 (c). The nine diagonal band matrix of the coefficients has the following structure:

$$\begin{bmatrix} \ddots & & & & & & & & \\ \ddots & \ddots & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & \ddots & \ddots & & & & & \\ & & & \ddots & \ddots & & & & \\ & & & & \ddots & \ddots & & & \\ & & & & & \ddots & \ddots & & \\ & & & & & & \ddots & \ddots & \\ & & & & & & & \ddots & \ddots \end{bmatrix}$$

A_{SW} A_S A_{SE} A_W A_P A_E A_{NW} A_N A_{NE}

Appendix D

Fourier transforms

Following Press et al. [56] for a continuous function of one variable $f(x) \in L^2(\mathbb{R})$, the *continuous* Fourier transform is defined as

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi kx} dx \quad (\text{D.1})$$

where $\xi \in \mathbb{R}$, the inverse *continuous* Fourier transform is defined as

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{i2\pi kx} dk \quad (\text{D.2})$$

Consider a complex periodic series $f(n)$ with N samples $(f_0, f_1, f_2, \dots, f_{N-1})$. The forward

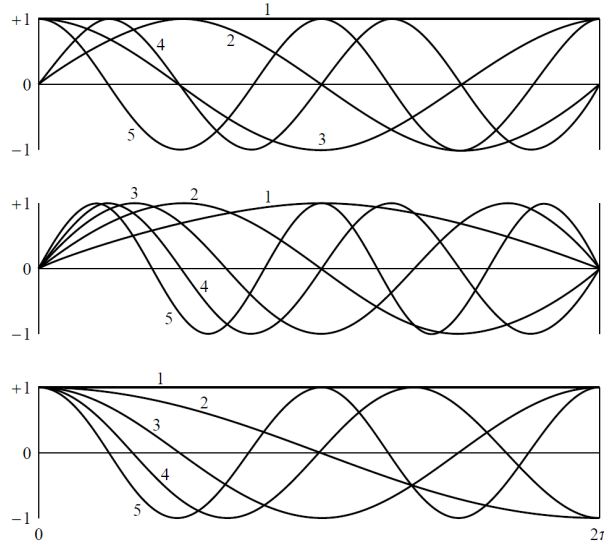


Figure D.1: (top) The trigonometric basis functions for a *complex* FFT of a periodic function. (center) The trigonometric basis functions for a *sine* FFT of a function with homogeneous Dirichlet boundary conditions. (bottom) The trigonometric basis functions for a *cosine* FFT of a function with homogeneous Neumann boundary conditions. Picture from [56].

discrete *complex* Fourier transform is defined as

$$\hat{f}_k = \sum_{n=0}^{N-1} f_n e^{-i2\pi kn/N} \quad \text{for } k = 0, 1, \dots, N-1 \quad (\text{D.3})$$

where $k \in \mathbb{Z}$. The trigonometric basis functions for a complex FFT of a periodic function is illustrated in Fig. D.1 (top). The inverse discrete Fourier transform is defined as

$$f_n = \frac{1}{N} \sum_{k=0}^{N-1} \hat{f}_k e^{i2\pi kn/N} \quad \text{for } n = 0, 1, \dots, N-1 \quad (\text{D.4})$$

For the real data with zero value at the boundaries (homogeneous Dirichlet boundary condition), the natural discrete Fourier transform to use is the *sine* transform (see Fig. D.1, center), given by

$$\hat{f}_k = \sum_{n=0}^{N-1} f_n \sin(\pi kn/N) \quad \text{for } k = 0, 1, \dots, N-1 \quad (\text{D.5})$$

and the inverse *sine* transform is given by

$$f_n = \frac{2}{N} \sum_{k=0}^{N-1} \hat{f}_k \sin(\pi kn/N) \quad \text{for } n = 0, 1, \dots, N-1 \quad (\text{D.6})$$

The other common boundary condition for differential equations is that the derivative of the function is zero at the boundaries (homogenous Neumann boundary condition). In the case of collocated arrangement of real data the natural discrete transform is the *cosine* transform (see Fig. D.1, bottom). The first form of the *cosine* transform uses $N+1$ points :

$$\hat{f}_k = \frac{f_0 + (-1)^k f_N}{2} + \sum_{n=1}^{N-1} f_n \cos(\pi kn/N) \quad \text{for } k = 0, 1, \dots, N \quad (\text{D.7})$$

the inverse *cosine* transform is defined as

$$f_n = \frac{2}{N} \left[\frac{\hat{f}_0 + (-1)^n \hat{f}_N}{2} + \sum_{k=1}^{N-1} \hat{f}_k \cos(\pi kn/N) \right] \quad \text{for } n = 0, 1, \dots, N \quad (\text{D.8})$$

In the case of staggered arrangement of real data (with homogeneous Dirichlet boundary conditions) the discrete *cosine* transform is defined as

$$\hat{f}_k = \sum_{n=0}^{N-1} f_{n+1/2} \cos [\pi k(n+1/2)/N] \quad \text{for } k = 0, 1, \dots, N-1 \quad (\text{D.9})$$

and the inverse *cosine* transform is defined as

$$f_{n+1/2} = \frac{2}{N} \left[\frac{\hat{f}_0}{2} + \sum_{k=1}^{N-1} \hat{f}_k \cos [\pi k(n+1/2)/N] \right] \quad \text{for } n = 0, 1, \dots, N-1 \quad (\text{D.10})$$

Appendix E

Turbulent structure identification criteria

Perhaps the most significant contribution of DNS to date has been the identification and eduction of turbulent structures. Since DNS supplies the pressure and velocity fields and velocity components gradients at each point in space and time, it allows the investigation of the relationship between the vortical structures with other quantities. Several methods can be used to visualize the coherent structures (eddies) in a turbulent flow. The pressure is effective in identifying the regions of strong rotation in vortex cores in a field without strong pressure gradient. Hunt et al. [47] proposed to use the Q criterion which is the second invariant of the deformation (or velocity gradient) tensor,

$$D_{ij} = \frac{\partial u_i}{\partial x_j} = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{pmatrix} \quad (\text{E.1})$$

Q is defined as

$$Q = \frac{1}{2}(\Omega_{ij}\Omega_{ij} - S_{ij}S_{ij}) = -\frac{1}{2}\frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i} \quad (\text{E.2})$$

where $S_{ij} = (u_{i,j} + u_{j,i})/2$ is the strain rate tensor which is the symmetric part of the deformation tensor and $\Omega_{ij} = (u_{i,j} - u_{j,i})/2$ is the rotation tensor which is the anti-symmetric part of the deformation tensor

$$D_{ij} = S_{ij} + \Omega_{ij}. \quad (\text{E.3})$$

The expanded expression for Q is given by

$$Q = v_y w_z + u_x v_y + u_x w_z - v_z w_y - v_x u_y - u_z w_x$$

In regions where $Q > 0$ vorticity is significant, and rotation dominates over shear. The Q criteria for two dimensions is presented by Weiss [50]. The vorticity norm can also be used as a criterion for coherent structure visualization. Vorticity is the curl of the velocity field

$$\vec{\omega} = \nabla \times \mathbf{u} = (w_y - v_z)\hat{\mathbf{i}} + (u_z - w_x)\hat{\mathbf{j}} + (v_x - u_y)\hat{\mathbf{k}}$$

its norm is defined as

$$|\vec{\omega}| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2} \quad (\text{E.4})$$

For more details we refer to McWilliams [43], Vincent and Meneguzzi [53] and Haller [101].

Appendix F

Direct solvers for linear systems with diagonal matrix of coefficients

Linear systems of equations are commonly encountered in implicit compact methods. For solving tridiagonal and pentadiagonal linear systems of equations, of the form; $[A]X = [B]$, the following algorithms can be used respectively. Both of them are direct methods based on complete lower-upper (LU) decomposition of the coefficient matrices. This is accomplished by forward elimination of lower diagonals and then backward substitution for the solution.

```
SUBROUTINE TRID (ibeg, iend, a, b, c, f)
! Matrix of the coefficients has 3 bands, (a b c), with b is the main diagonal,
! a is the lower diagonal, c is the upper diagonal and f is used to store the right hand side.
! The solution vector will come back in f.
IMPLICIT NONE
INTEGER i, ibeg, iend
REAL(8) z
REAL(8), DIMENSION (ibeg:iend) :: a, b, c, f

f(ibeg) = f(ibeg)/b(ibeg)
c(ibeg) = c(ibeg)/b(ibeg)
DO i=ibeg+1,iend-1
  z = 1./(b(i) - a(i)*c(i-1))
  c(i) = c(i)*z
  f(i) = (f(i) - a(i)*f(i-1))*z
END DO

f(iend) = (f(iend) - a(iend)*f(iend-1))/(b(iend) - a(iend)*c(iend-1))
DO i=iend-1,ibeg,-1
  f(i) = f(i) - c(i)*f(i+1)
END DO

RETURN
END SUBROUTINE TRID
```

```
*****
```

```
SUBROUTINE PENTAD (N, E, A, D, C, F, B)
```

```
! Akron University - Mathematics department
```

```
! Matrix has 5 bands, E A D C F, with D being the main diagonal,
```

```
! E and A are the lower diagonals, C and F are the upper diagonals.
```

```
! E is defined for rows i = 3:N, but in the code it is defined as E(1) to E(N-2)
```

```
! A is defined for rows i = 2:N, but in the code it is defined as A(1) to A(N-1)
```

```
! D is defined for rows i = 1:N
```

```
! C is defined for rows i = 1:N-1, but in the code the last element is not used
```

```
! F is defined for rows i = 1:N-2, but in the code the last 2 elements are not used
```

```
! B is the right-hand side
```

```
! The solution vector will come back in E
```

```
IMPLICIT NONE
```

```
INTEGER I, N
```

```
REAL(8), DIMENSION (1:N) :: E, A, D, C, F, B
```

```
REAL(8) XMULT
```

```
DO I = 2, N-1
```

```
XMULT = A(I-1)/D(I-1)
```

```
D(I) = D(I) - XMULT*C(I-1)
```

```
C(I) = C(I) - XMULT*F(I-1)
```

```
B(I) = B(I) - XMULT*B(I-1)
```

```
XMULT = E(I-1)/D(I-1)
```

```
A(I) = A(I) - XMULT*C(I-1)
```

```
D(I+1) = D(I+1) - XMULT*F(I-1)
```

```
B(I+1) = B(I+1) - XMULT*B(I-1)
```

```
ENDDO
```

```
XMULT = A(N-1)/D(N-1)
```

```
D(N) = D(N) - XMULT*C(N-1)
```

```
E(N) = (B(N) - XMULT*B(N-1))/D(N)
```

```
E(N-1) = (B(N-1) - C(N-1)*E(N)) / D(N-1)
```

```
DO I = N-2,1,-1
```

```
E(I) = (B(I) - F(I)*E(I+2) - C(I)*E(I+1)) / D(I)
```

```
ENDDO
```

```
RETURN
```

```
END SUBROUTINE PENTAD
```

```
*****
```

Bibliography

- [1] S. W. R. Hamilton. Lectures on quaternions. *Hodges and Smith*, 1853.
- [2] V. Strouhal, Über eine besondere Art der Tonerregung [On a particular way of sound generation]. *Annalen der Physik und Chemie*, Band 241, 216-251, 1878.
- [3] T. von Kármán, Über den Mechanismus des Widerstandes, den ein bewegter Körper in einer Flüssigkeit erfährt [About the mechanism of resistance that a moving body undergoes in a liquid]. *Nachrichten der K. Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-physikalische Klasse*, 509-517, 1911.
- [4] L. Rayleigh, Aeolian tones. *Philosophical Magazine Letters*, Vol. 29, 433-444, 1915.
- [5] G. I. Taylor, Stability of a viscous liquid contained between two rotating cylinders. *Philosophical Transactions of the Royal Society A*, Vol. 223, 289-343, 1923.
- [6] C. M. Breder, The locomotion of fishes. *Zoologica*, Vol. 4, 159-297, 1926.
- [7] H. Bénard, Sur les écarts des valeurs de la fréquence des tourbillons alternés par rapport à la loi de similitude dynamique. *Comptes Rendus de l'Académie des Sciences, Paris*, Vol. 183, 20-22, 1926.
- [8] A. Thom, The flow past circular cylinders at low speeds. *Proceedings of the Royal Society, London, Section A*, Vol. 141, 651-669, 1933.
- [9] H. C. Brinkmann, A calculation of the viscous force exerted by a flowing fluid on a swarm of particles. *Applied Science Research*, Vol. 1, 27-34, 1947.
- [10] L. S. G. Kovasznay, Hot-wire investigation of the wake behind cylinders at low Reynolds numbers. *Proceedings of the Royal Society, London, A (Mathematical, Physical & Engineering Sciences)*, 198, 1949.
- [11] L. C. Woods, A note on the numerical solution of fourth-order differential equations. *Aero Quarterly Journal of the American Society of Aviation Artists*, Vol. 5, 176-182, 1954.
- [12] A. Roshko, On the Development of Turbulent Wakes From Vortex Streets. *NACA Reports*, No. 1191, 1954.
- [13] R. G. Brown, Exponential smoothing for predicting demand. *Cambridge, Massachusetts: Arthur D. Little Inc.*, 15-, 1956.
- [14] C. C. Holt, Forecasting trends and seasonal by exponentially weighted averages. *Office of Naval Research Memorandum, Carnegie Institute of Technology* 52, 1957.

- [15] F. G. Shuman, Numerical methods in weather prediction: II Smoothing and filtering. *Mon. Wea. Rev.*, Vol. 85, 357-361, 1957.
- [16] VG. Jensen, Viscous flow round a sphere at low Reynolds numbers (≤ 40). *Proceedings of the Royal Society, London A*, Vol. 294, 346-366, 1959.
- [17] L. Collatz, The Numerical Treatment of Differential Equations. *Springer-Verlag, Berlin*, 1960.
- [18] R. Wille, Karman Vortex Streets. *Advances in Applied Mechanics, Academic, New York*, Vol. 6, 273-287, 1960.
- [19] P. R. Winters, Forecasting sales by exponentially weighted moving averages. *Management Science*, Vol. 6, 324-342, 1960.
- [20] F. H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, Vol. 8, No. 12, 2182-2189, 1965.
- [21] A. J. Chorin, A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, Vol. 2, 12-26, 1967.
- [22] A. J. Chorin, Numerical solution of the Navier–Stokes equations. *Mathematics of Computation*, Vol. 22, 745-762, 1968.
- [23] R. Temam, Une méthode d'approximation des solutions des équations Navier–Stokes. *Bull. Soc. Math. France*, Vol. 98, 115-152, 1968.
- [24] D. C. Thoman and A. A. Szewczyk, Time-dependent viscous flow over a circular cylinder. *Physics of Fluids Suppl. II*, 79-86, 1969.
- [25] M. J. Lighthill, Hydromechanics of aquatic animal propulsion. *Annual Review of Fluid Mechanics*, Vol. 1(1), 413-446, 1969.
- [26] M. J. Lighthill, Aquatic animal propulsion of high hydromechanical efficiency. *Journal of Fluid Mechanics*, Vol. 44, 265-301, 1970.
- [27] A. S. Monin and A. M. Yaglom, Statistical Fluid Mechanics, *MIT Press, Cambridge, MA*, 1971.
- [28] M. J. Lighthill, Large-amplitude elongated-body theory of fish locomotion. *Proceedings of the Royal Society, London, Mech. B*, Vol. 179, 125-138, 1971.
- [29] W. R. Briley, A numerical study of laminar separation bubbles using the Navier–Stokes equations. *Journal of Fluid Mechanics*, Vol. 47, 713-736, 1971.
- [30] P. Roache, Computational Fluid Dynamics. *Hermosa Press*, 1972.
- [31] E. Berger and R. Wille, Periodic flow phenomena. *Annual Review of Fluid Mechanics*, Vol. 4, 313-340, 1972.
- [32] S.A. Orszag and M. Israeli, Numerical simulation of viscous incompressible flows, *Annual Review of Fluid Mechanics*, Vol. 6, 281-318, 1974.

- [33] R. S. Hirsh, High order accurate difference solutions of fluid mechanics problems by a compact differencing technique. *Journal of Computational Physics*, Vol. 19, 90-109, 1975.
- [34] C. L. Lin, D. W. Pepper and S. C. Lee, Numerical methods for separated flow solutions around a circular cylinder. *AIAA Journal*, Vol. 14, No. 7, 900-907, 1976.
- [35] P. Bontoux, B. Forestier and B. Roux, Analyse et optimisation d'une méthode de haute précision pour la résolution des équations de Navier-Stokes instationnaires. *Journal de Mécanique Appliquée*, Vol. 2, no. 3, 291-316, 1978.
- [36] B. Roux, P. Bontoux, T. P. Loc and O. Daube, Optimisation of Hermitian methods for Navier-Stokes equations in the vorticity and stream-function formulation. *West Germany, September 9-15, 1979. (A81-36526 16-34) Berlin, Springer-Verlag*, p. 450-468, 1980.
- [37] G. Martinez, Caractéristiques dynamiques et thermiques de l'écoulements autour d'un cylindre circulaire à nombres de Reynolds modérés. *Thèse D.I.-I.N.P. Toulouse*, 1979.
- [38] R. H. Kraichnan and D. Montgomery, Two-dimensional turbulence. *Reports on Progress in Physics*, Vol. 43, 1980.
- [39] C. Rhie, W. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, Vol. 21, 1525-1532, 1983.
- [40] E. Arquis et J.P. Caltagirone, Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluid - milieux poreux: application à la convection naturelle. *Comptes Rendus de l'Académie des Sciences, Paris*, 2(299), 1-4, 1984.
- [41] J. J. Videler and F. Hess, Fast continuous swimming of two pelagic predators, saithe (pollachius virens) and mackerel (scomber scombrus): A kinematic analysis. *Journal of experimental biology*, Vol. 109, 209-228, 1984.
- [42] Y. Lecointe and J. Piquet, On the use of several compact methods for the study of unsteady incompressible viscous flow around a circular cylinder. *Computers & Fluids*, Vol. 12, No. 4, 255-280, 1984.
- [43] J. McWilliams, The emergence of isolated coherent vortices in turbulent flow. *Journal of Fluid Mechanics*, Vol. 146, 21-43, 1984.
- [44] J. Kim and P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, Vol. 59, 308-323, 1985.
- [45] J. C. Simo, A finite strain beam formulation. The three-dimensional dynamic problem. Part I, *Computer Methods in Applied Mechanics and Engineering*, Vol. 49, 55-70, 1985.
- [46] M. Rosenfeld, D. Kwak and M. Vinokur, A solution method for the unsteady and incompressible Navier-Stokes equations in generalized coordinate systems. *AIAA Paper*, Vol. 88, 718-, 1988.
- [47] J. C. R. Hunt, A. Wray and P. Moin, Eddies, stream, and convergence zones in turbulent flows. *Center for Turbulence Research Report, CTR-S88*, 1988.

- [48] S. E. Rogers and D. Kwak, An upwind differencing scheme for the time-accurate incompressible Navier–Stokes equations. *AIAA Journal*, Vol. 28, No. 2, 253-262, 1990.
- [49] J. Liandrat and P. Tchamitchian, Resolution of the 1D regularized Burgers equation using a spatial wavelet approximation. Technical Report NASA Contractor Report 187480, NASA Langley Research Center, Hampton VA, 1990.
- [50] J. Weiss, The dynamics of enstrophy transfer in 2-dimensional hydrodynamics. *Physica D*, Vol. 48, 273–294, 1991.
- [51] M. S. Triantafyllou, G. S. Triantafyllou, R. Gopalkrishnan, Wake mechanics for thrust generation in oscillating foils. *Physics of Fluids A*, Vol. 3, 2835-2837, 1991.
- [52] P. M. Gresho, Incompressible fluid dynamics: some fundamental formulation issues. *Annual Review of Fluid Mechanics*, Vol. 23, 413–53, 1991 .
- [53] A. Vincent and M. Meneguzzi, The spatial structure and statistical properties of homogeneous turbulence. *Journal of Fluid Mechanics*, Vol. 225, 1–20, 1991.
- [54] D. L. Donoho, Interpolating wavelet transforms. Technical Report 408, Department of Statistics, Stanford University, 1992.
- [55] S. K. Lele, Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, Vol. 103, 16-42, 1992.
- [56] H. Press, A. Teukolsky, T. Vetterling and P. Flannery, Numerical Recipes in Fortran 77 : The Art of Scientific Computing. 2nd Edition, Cambridge University Press, ISBN: 0-521-43064-X, 1992.
- [57] T. Y. Hou and B. T. R. Wetton, Convergence of a finite difference scheme for the Navier–Stokes equations using vorticity boundary conditions. *SIAM Journal on Numerical Analysis*, Vol. 29, 615-639, 1992.
- [58] T. Miyake, Y. Sakamoto, H. Tokunaga and N. Satofuka, Numerical solution of incompressible flow using two-step, one-stage Runge–Kutta time integration scheme. Presented at the 1st European Computational Fluid Dynamics Conference, Brussels, Belgium, 7-11 September, 1992.
- [59] G. S. Triantafyllou, M. S. Triantafyllou, M. A. Grosenbaugh, Optimal thrust development in oscillating foils with application to fish propulsion. *Journal of Fluids and Structures*, Vol. 7, 205-224, 1993.
- [60] A. Belov, L. Martinelli and A. Jameson, A new implicit algorithm with multigrid for unsteady incompressible flow calculations. Presented at the 33rd Aerospace Sciences Meeting and Exhibit, AIAA Paper 95-0049, 9-12 January, 1995.
- [61] W. F. Spitz and G. F. Carey, High-order compact scheme for the steady stream-function vorticity equations. *International Journal for Numerical Methods in Engineering*, Vol. 38, 3497-3512, 1995.
- [62] E. Weinan and L. Jian-Guo, Vorticity boundary condition and related issues for finite difference schemes. *Journal of Computational Physics*, Vol. 124(2), 368-382, 1996.

- [63] E. Hairer and G. Wanner, Solving ordinary differential equations II, stiff and differential-algebraic problems, *Springer*, 1996.
- [64] B. L. Bihari and A. Harten, Multiresolution schemes for the numerical solution of 2-D conservation laws. *SIAM Journal on Scientific Computing*, Vol. 18, No. 2, 315-354, 1997.
- [65] C. H. K. Williamson and Roshko, Vortex formation in the wake of an oscillating cylinder. *Journal of Fluids and Structure*, Vol. 2, 355-381, 1988.
- [66] P. Moin and K. Mahesh, Direct Numerical Simulation: A Tool in Turbulence Research. *Annual Review of Fluid Mechanics*, Vol. 30, 539-578, 1998.
- [67] J. Carling, T. L. Williams and G. Bowtell, Self-propelled anguilliform swimming: simultaneous solution of the two-dimensional Navier–Stokes equations and Newton’s laws of motion. *Journal of Experimental Biology*, Vol. 201, 3143-3166, 1998.
- [68] C. Liu, X. Zheng and C. Sung, Preconditioned multigrid methods for unsteady incompressible flows. *Journal of Computational Physics*, Vol. 139, 35-57, 1998.
- [69] Y. Morinishi, T. S. Lund, O. V. Vasilyev and P. Moin, Fully conservative higher order finite difference schemes for incompressible flow. *Journal of Computational Physics*, Vol. 143, 90-124, 1998.
- [70] J. M. Powers and S. Paolucci, Manifold methods for energetic materials. Presented at the IMA workshop on High-Speed Combustion in Gaseous and Condensed-Phase Energetic Materials, University of Minnesota, 11 November 1999.
- [71] D. Barrett, M. S. Triantafyllou, D. Yue, M. Grosenbauch and M. Wolfgang, Drag reduction in fish-like locomotion. *Journal of Fluid Mechanics*, Vol. 392, 182-212, 1999.
- [72] M. Napolitano, G. Pascasio and L. Quartapelle, A review of vorticity conditions in the numerical solution of the $\zeta - \psi$ equations. *Computers & Fluids*, Vol. 28, 139-185, 1999.
- [73] P. Angot, C.-H. Bruneau and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, Vol. 81, no. 4, 497-520, 1999.
- [74] A. T. Johnson and V. C. Patel, Flow past a sphere up to a Reynolds number of 300. *Journal of Fluid Mechanics* 378, 19-70, 1999.
- [75] K. Khadra, P. Angot, S. Parniex and J. P. Caltagirone. Fictitious domain approach for numerical modelling of Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, Vol. 34, 651-684, 2000.
- [76] M. C. Lai and C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, Vol. 160, 705-719, 2000.
- [77] P. Orlandi, Fluid flow phenomena: A numerical toolkit. *Springer*, 2000.

- [78] C. Farhat and M. Lesoinne, Two efficient staggered procedures for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, Vol. 182, 499-516, 2000.
- [79] M. Forestier, Etude par méthode spectrale de sillages tridimensionnels en fluide stratifié. *Thèse de Doctorat, Université de Nice-Sophia*, 2000.
- [80] O. V. Vasilyev, High order finite difference schemes on non-uniform meshes with good conservation properties. *Journal of Computational Physics*, Vol. 157, 746-761, 2000.
- [81] U. Piomelli, Large-eddy and direct simulation of turbulent flows, *Department of mechanical engineering, University of Maryland*, 2001.
- [82] A. G. Malan and R. W. Lewis, An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows. *International Journal for Numerical Methods in Engineering*, Vol. 54, 695-714, 2002.
- [83] Q. Zhu, M. Wolfgang, D. Yue and M. S. Triantafyllou, Three-dimensional flow structures and vorticity control in fish-like swimming. *Journal of Fluid Mechanics*, Vol. 468, 1-28, 2002.
- [84] C. Wang and J. G. Liu, Analysis of finite difference schemes for unsteady Navier–Stokes equations in vorticity formulation. *Numerische Mathematik*, Vol. 91, 543-576, 2002.
- [85] M. R. Visbal and D. V. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, *Journal of Computational Physics*, Vol. 181, 155-185, 2002.
- [86] H. Johnston, J.G. Liu, Finite difference schemes for incompressible flow based on local pressure boundary conditions, *Journal of Computational Physics*, Vol. 180, 120-154, 2002.
- [87] H. J. H. Clercx and G. J. F. van Heijst, Dissipation of kinetic energy in two-dimensional bounded flows. *Physical Review E*, Vol. 65, 066305, 2002.
- [88] P. A. Durbin and G. Iaccarino, An approach to local refinement of structure grids. *Journal of Computational Physics*, Vol. 181, Issue 2, 639-653, 2002.
- [89] O. Roussel, Développement d'un algorithme multirésolution adaptatif tridimensionnel pour les équations aux dérivées partielles paraboliques. Application l'étude des instabilités thermo-diffusives de flamme. PhD thesis, Université d'Aix-Marseille, 2003.
- [90] N. Hansen, S. D. Müller and P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computations*, Vol. 11 (1), 1-18, 2003.
- [91] A. Gilmanov, F. Sotiropoulos and E. Balaras, A general reconstruction algorithm for simulating flows with 3D immersed boundaries on Cartesian grids. *Journal of Computational Physics*, Vol. 191, 660-669, 2003.
- [92] G. Carbou and P. Fabrie, Boundary layer for a penalization method for viscous incompressible flow. *Advances in Differential Equations*, Vol. 8, 1453-, 2003.

- [93] C. H. Tai and Y. Zhao, Parallel unsteady incompressible viscous flow computations using an unstructured multigrid method. *Journal of Computational Physics*, Vol. 192, 277-311, 2003.
- [94] J. D. Hawkins, C. A. Sepulveda, J. B. Graham and K. A. Dickson, Swimming performance studies on the eastern Pacific bonito *Sarda chiliensis*, a close relative of the tunas (family Scombridae) part II. Kinematics. *The Journal of Experimental Biology*, Vol. 206, 2749-2758, 2003.
- [95] E. Krishnan, L^AT_EX Tutorials. *Indian T_EX Users Group*, 2003.
- [96] P. A. Davidson, Turbulence: An introduction for scientists and engineers. *Oxford University Press*, 2004.
- [97] S. Abide and S. Viazzo, A 2D compact fourth-order projection decomposition method. *Journal of Computational Physics*, Vol. 206, 252-276, 2005.
- [98] M. N. Linnick and H. F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*, Vol. 204, 157-192, 2005.
- [99] R. Mittal and G. Iaccarino, Immersed boundary methods. *Annual Reviews of Fluid Mechanics*, Vol. 37, 239-261, 2005.
- [100] A. Leroyer and M. Visonneau, Numerical methods for RANSE simulations of a self-propelled fish-like body. *Journal of Fluids and Structures*, Vol. 20, 975-991, 2005.
- [101] G. Haller, An objective definition of a vortex. *Journal of Fluid Mechanics*, Vol. 525, 1-26, 2005.
- [102] K. Schneider and M. Farge, Decaying two-dimensional turbulence in a circular container. *Physical Review Letters*, Vol. 95, 244502, 2005.
- [103] D. Rempfer, On boundary conditions for incompressible Navier-Stokes problems. *Applied Mechanics Reviews*, Vol. 59, Issue 3, 107-125, 2006.
- [104] H. J. H. Clercx and C.-H. Bruneau, The normal and oblique collision of a dipole with a no-slip boundary. *Computers and Fluids*, Vol. 35, 245, 2006.
- [105] S. Kern and P. Koumoutsakos, Simulations of optimized anguilliform swimming. *Journal of Experimental Biology*, Vol. 209, 4841-4857, 2006.
- [106] F. Boyer, M. Porez and W. Khalil, Macro-continuous computed torque algorithm for a three-dimensional eel-like robot. *IEEE Transactions on Robotics*, Vol. 22, No. 4, 763-775, August 2006.
- [107] K. Schneider and M. Farge, Wavelets: Mathematical Theory. *Encyclopedia of Mathematical Physics*, Elsevier, 426-438, 2006.
- [108] J. L. Guermond, P. Mineev and J. Shen, An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, 6011-6045, 2006.

- [109] A. A. Siddiqui and M. T. Mustafa, Wavelet optimized finite difference method with non-static re-gridding. *Applied Math. and Computation*, Vol. 186, 203-211, 2007.
- [110] J. Deng, X. M. Shao and Z. S. Yu, Hydrodynamic studies on two traveling wavy foils in tandem arrangement. *Physics of Fluids*, Vol. 19, 113104, 2007.
- [111] J. W. Kim, Optimised boundary compact finite difference schemes for computational aeroacoustics. *Journal of Computational Physics*, Vol. 225, 995-1019, 2007.
- [112] G. H. Keetels, U. D'Ortona, W. Kramer, H. J. H. Clercx, K. Schneider and G. J. F. van Heijst, Fourier spectral and wavelet solvers for the incompressible Navier–Stokes equations with volume-penalization: Convergence of a dipole-wall collision. *Journal of Computational Physics*, Vol. 227, 919-945, 2007.
- [113] W. Kramer, H. J. H. Clercx and G. J. F. van Heijst, Vorticity dynamics of a dipole colliding with a no-slip wall. *Physics of Fluids*, Vol. 19, 126603, 2007.
- [114] J. M. McDonough, Lectures in computational fluid dynamics of incompressible flow: Mathematics, algorithms and implementations. *Departments of mechanical engineering and mathematics, University of Kentucky*, 2007.
- [115] J. M. McDonough, Introductory lectures on turbulence Physics: Mathematics and Modeling. *Departments of mechanical engineering and mathematics, University of Kentucky*, 2007.
- [116] M. V. Pham, F. Plourde and S. Doan Kim, Strip decomposition parallelization of fast direct Poisson solver on a 3D Cartesian staggered grid. *International Journal of Computer Science and Engineering (IJCSE)*, Vol. 1, No. 1, 31-40, 2007.
- [117] K. Taira and T. Colonius, The immersed boundary method: a projection approach. *Journal of Computational Physics*, Vol. 225, 2118-2137, 2007.
- [118] M. Farge, Numerical experimentation: A third way of study nature. *Frontiers of Computational Science, Springer*, 15-28, 2007.
- [119] K. Namkoong, J.Y. Yoo, H.G. Choi, Numerical analysis of two-dimensional motion of a freely falling circular cylinder in an infinite fluid, *Journal of Fluid Mechanics*, Vol. 604, 33-53, 2008.
- [120] M. El Rafei, M. Alamir, N. Marchand, M. Porez and F. Boyer, Multi-variable constrained control approach for a three-dimensional Eel-like robot. *IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems, IROS*, Nice, France, 2008.
- [121] R. Pasquetti, R. Bwemba and L. Cousin, A pseudo-penalization method for high Reynolds number unsteady flows. *Applied Numerical Mathematics*, Vol. 58, 946-954, 2008.
- [122] M. Minguez, Simulations des grandes échelles de haute précision d'écoulements turbulents autour de géométries complexes : application à un modèle de véhicule automobile [Large eddy simulation of turbulent flows over complex geometries: Application to a simplified car model]. *PhD thesis, Université Paul Cézanne, Aix-Marseille III*, 2008.

- [123] M. Coquerelle and G.-H. Cottet, A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *Journal of Computational Physics*, Vol. 227, 9121-9137, 2008.
- [124] M. Valizadeh, F. Sabetghadam, A. Mostofi and M. Salari, Implicit steady and unsteady solutions to incompressible Navier–Stokes equations based on artificial compressibility method in general curvilinear coordinate. *The 11th fluid dynamics conference, Khaje Nassir Addin Toussi university, Tehran*, 1775-, 2008.
- [125] P. Parnaudeau, J. Carlier, D. Heitz, and E. Lamballais, Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900. *Phys. Fluids*, Vol. 20:085101, 2008.
- [126] D. Kolomenskiy and K. Schneider. A Fourier spectral method for the Navier–Stokes equations with volume penalization for moving solid obstacles. *Journal of Computational Physics*, Vol. 228, no. 16, 5687-5709, 2009.
- [127] F. Sabetghadam, S. A. Ghaffari and M. Dadashi, Implementation of vortex stretching into the two-dimensional Navier–Stokes equations via arbitrary external straining. *Advances in Turbulence XII, Springer Proceedings in Physics*, 132, 2009.
- [128] T. Schnipper, A. Andersen and T. Bohr, Vortex wakes of flapping foil. *Journal of Fluid Mechanics*, Vol. 633, 411-423, 2009.
- [129] S. Laizet and E. Lamballais, High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *Journal of Computational Physics*, Vol. 228, 5989-6015, 2009.
- [130] K. Brix, S. S. Melian, S. Müller and G. Schieffer, Parallelisation of multiscale-based grid adaptation using space-filling curves. *ESAIM: Proceedings*, Vol. 29, 108-129, 2009.
- [131] S. Laizet, E. Lamballais and J.C. Vassilicos. A numerical strategy to combine high-order schemes, complex geometry and parallel computing for high resolution DNS of fractal generated turbulence. *Computers & Fluids*, Vol. 39(3), 471-484, 2010.
- [132] N. Li and S. Laizet, 2DECOMPFFT a highly scalable 2d decomposition library and FFT interface. *In Cray User Group, Edinburgh*, 2010.
- [133] A. Paknejad. Development of an aggregation based algebraic multi-grid method for numerical solution of incompressible flows. *Master thesis, MUT*, 2010.
- [134] K.S. Yeo, S.J. Ang and C. Shu, Simulation of fish swimming and manoeuvring by an SVD-GFD method on a hybrid meshfree-Cartesian grid. *Computers & Fluids*, Vol. 39, 403-430, 2010.
- [135] D. Rossinelli, M. Bergdorf, G. H. Cottet and P. Koumoutsakos, GPU accelerated simulations of bluff body flows using vortex particle methods. *Journal of Computational Physics*, Vol. 229, 3316-3333, 2010.
- [136] B. Hejazialhosseini, D. Rossinelli, M. Bergdorf and P. Koumoutsakos, High order finite volume methods on wavelet-adapted grids with local time-stepping on multicore architectures for the simulation of shock-bubble interactions. *Journal of Computational Physics*, Vol. 229, 8364-8383, 2010.

- [137] S. M. Sachs, D. C. Sternel and M. Schäfer, Implicit partitioned coupling with global multigrid in fluid–structure interaction. *V European Conference on Computational Fluid Dynamics - ECCOMAS CFD - Lissabon*, [Conference or Workshop Item], 2010.
- [138] Shashank , J. Larsson and G. Iaccarino, A co-located incompressible Navier–Stokes solver with exact mass, momentum and kinetic energy conservation in the inviscid limit. *Journal of Computational Physics*, Vol. 229, 4425–4430, 2010.
- [139] M. Belliarda and C. Fournier, Penalized direct forcing and projection schemes for Navier–Stokes. *Comptes Rendus Mathématique de l’Académie des Sciences, Paris*, Vol. 348, Issues 19–20, 1133–1136, 2010.
- [140] D. Kolomenskiy and K. Schneider, Numerical simulations of falling leaves using a pseudo-spectral method with volume penalization. *Theoretical and Computational Fluid Dynamics*, Vol. 24(1-4), 169–173, 2010.
- [141] K. Schneider and O. V. Vasilyev, Wavelet methods in computational fluid dynamics. *Annual Review of Fluid Mechanics*, Vol. 42, 473–503, 2010.
- [142] S. Laizet and N. Li, Incompact3d, a powerful tool to tackle turbulence problems with up to $o(10^5)$ computational cores. *Int. J. Numer. Methods Fluids*, 67(11):1735–1757, 2011.
- [143] S. A. Ghaffari, Development of an adaptive multiresolution method to study the near wall behavior of two-dimensional vortical flows. Master thesis, Mécanique des Fluides et Physique Non-linéaire, *Aix-Marseille University*, 2011.
- [144] M. Gazzola, P. Chatelain, W. M. van Rees and P. Koumoutsakos, Simulations of single and multiple swimmers with non-divergence free deforming geometries. *Journal of Computational Physics*, Vol. 230, 7093–7114, 2011.
- [145] M. Bergmann and A. Iollo, Modeling and simulation of fish-like swimming. *Journal of Computational Physics*, Vol. 230, 329–348, 2011.
- [146] B. J. Boersma, A 6th order staggered compact finite difference method for the incompressible Navier–Stokes and scalar transport equations. *Journal of Computational Physics*, Vol. 230, Issue 12, 4940–4954, 2011.
- [147] J. H. Seo and R. Mittal, A high-order immersed boundary method for acoustic wave scattering and low-Mach number flow-induced sound in complex geometries. *Journal of Computational Physics*, Vol. 230, 1000–1019, 2011.
- [148] G. Bonfigli, High-order finite-difference implementation of the immersed-boundary technique for incompressible flows. *Computers & Fluids*, Vol. 46, 2–11, 2011.
- [149] S. Wang and X. Zhang, An immersed boundary method based on discrete stream function formulation for two- and three-dimensional incompressible flows. *Journal of Computational Physics*, Vol. 230, 3479–3499, 2011.
- [150] E. Lamballais, V. Fortune and S. Laizet, Straightforward high-order numerical dissipation via the viscous term for Direct and Large Eddy Simulation. *Journal of Computational Physics*, Vol 230-9, pp 3270–3275, 2011.

- [151] R. Nguyen van yen, M. Farge and K. Schneider, Energy dissipating structures produced by walls in two-dimensional flows at vanishing viscosity. *Physical Review Letters*, Vol. 106, 184502, 2011.
- [152] M. Farge, Oh! Une idée, c'est si rare! *Wissenschaftskolleg zu Berlin, Yearbook 2009-2010*, 62-74, 2011.
- [153] H. Luo, H. Dai, P. J.S.A. Ferreira de Sousa and B. Yin, On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries. *Computers & Fluids*, Vol. 56, 61-76, 2012.
- [154] A. Lazarus, J. T. Miller, P. M. Reis, A quaternion-based continuation method to follow the equilibria and stability of slender elastic rods. *Published by arXiv:1212.5739v1*, 2012.
- [155] M. Gazzola, W. M. Van Rees and P. Koumoutsakos, C-start: optimal start of larval fish. *Journal of Fluid Mechanics*, Vol. 698, 5-18, 2012.
- [156] NIST/SEMATECH e-Handbook of Statistical Methods, 2012.
<http://www.itl.nist.gov/div898/handbook/>
- [157] C. Eloy, Optimal Strouhal number for swimming animals. *Journal of Fluids and Structures*, Vol. 30, 205-218, 2012.
- [158] A. Belkhiri, Modélisation dynamique de la locomotion compliant : Application au vol battant bio-inspiré de l'insecte. *Thèse de doctorat, École nationale supérieure des mines de Nantes*, 2013.
- [159] T. Engels, D. Kolomenskiy, K. Schneider and J. Sesterhenn. Two-dimensional simulation of the fluttering instability using a pseudospectral method with volume penalization. *Computers & Structures*, Vol. 122, 101-113, 2013.
- [160] W. M. Van Rees, M. Gazzola and P. Koumoutsakos, Optimal shapes for anguilliform swimmers at intermediate Reynolds numbers. *Journal of Fluid Mechanics Rapids*, Vol. 722, R3 1-12, 2013.
- [161] H. Emdad, A. R. Mostofizadeh and S. A. Mousavinia, Optimal control of vortex shedding around square cylinder via reduced order model. *Journal of mechanical engineering, Trbiat Modares University, Iran*, Vol 13, no. 5, 49-62 , 2013.
- [162] M. Gazzola, M. Argentina and L. Mahadevan, Scaling macroscopic aquatic locomotion. *Nature Physics*, Vol. 10, 758-761, 2014.
- [163] J. A. Morales, M. Leroy, W. J. T. Bos and K. Schneider, Simulation of confined magnetohydrodynamic flows with Dirichlet boundary conditions using a pseudo-spectral method with volume penalization. *Journal of Computational Physics*, Vol. 274, 64-94, 2014.
- [164] R. Nguyen van yen, D. Kolomenskiy and K. Schneider, Approximation of the Laplace and Stokes operators with Dirichlet boundary conditions through volume penalization: a spectral viewpoint. *Numerische Mathematik*, Vol. 128, 301-338, 2014.

- [165] Wikipedia contributors, Fish. *Wikipedia, The Free Encyclopedia*, 2014.
- [166] Wikipedia contributors, Fish locomotion. *Wikipedia, The Free Encyclopedia*, 2014.
- [167] F. Sotiropoulos and X. Yang, Immersed boundary methods for simulating fluid–structure interaction. *Progress in Aerospace Sciences*, Vol. 65, 1-21, 2014.
- [168] P. Bontoux, S. Viazzo, K. Schneider and S. A. Ghaffari, An efficient algorithm for simulation of forced deformable bodies interacting with incompressible flows; Application to fish swimming. ECCM V and 11th *World Congress on Computational Mechanics*, Vol. II, 787-798, Barcelona, Spain, 2014.
- [169] S. A. Ghaffari and K. Schneider, Development of an adaptive multiresolution method to study the near wall behavior of two-dimensional vortical flows. Research report, Laboratoire M2P2-UMR 7340-CNRS, Centrale Marseille and Aix-Marseille University, 2014, <http://hal.archives-ouvertes.fr/hal-00959469>.
- [170] S. A. Ghaffari, S. Viazzo, K. Schneider and P. Bontoux, Simulation of forced deformable bodies interacting with two-dimensional incompressible flows; Application to fish-like swimming. *International Journal of Heat and Fluid Flow*, Vol. 51, 88–109, 2015, <http://hal.archives-ouvertes.fr/hal-00967077>.
- [171] *The compact solver is developed in FORTRAN and is accessible to all by sending a mail to:* ghaffari@L3m.univ-mrs.fr or s.amin.ghaffary@gmail.com
- [172] *The MR solver is developed in FORTRAN and is accessible for all by sending a mail to:* s.amin.ghaffary@gmail.com or ghaffari@L3M.univ-mrs.fr